University of Alberta

Agent Embedded Simulation Modeling Framework for Construction Engineering and Management Applications

by

Anthony Allan Van Tol ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of
the requirements for the degree of Doctor of Philosophy

in

Construction Engineering and Management

Department of Civil and Environmental Engineering

Edmonton, Alberta

Spring 2005

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

0-494-08306-9

Your file *Votre référence*
ISBN
Our file *Notre reference*
ISBN

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

The presented research addresses concerns regarding the precision and proficiency of how current construction projects are planned, estimated, and executed. A new approach for construction engineering and management problem-solving is presented. This new problem-solving approach makes use of a simulation modeling environment with embedded intelligent agents to both capture the beneficial properties of simulation modeling with respect to the precision in which construction problems are represented and increase the proficiency with which those problems are solved.

The research focuses on the development of an agent structure that can be deployed within the simulation environment. The agent structure developed provides a means through which autonomous intelligent objects can be employed within the simulation modeling environment to assume roles and possess knowledge otherwise demanded by simulation modelers. The culmination of this development effort has resulted in a generic agent structure that can be instructed to assume a variety of roles and responsibilities during simulation runtime. The agent structure also possesses the flexibility to make use of varying intelligence facilities such as belief networks, feed-forward back propagation neural networks, algorithms, and rule bases. The second component of this research focuses on the development of an agent framework that allows several agents to be employed within the same simulation modeling environment with sufficient organization and governance to ensure that the total influence by the agents on the simulation model follows a common good.

Acknowledgements

The author would like to express his gratitude and appreciation to his supervisor Dr. S. M. AbouRizk for his support and guidance as well as Dr. Al-Hussein, Dr. Polikar, Dr. Flynn and Dr. Froese for their service on his evaluation committee.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION 1

1.1 Introduction 1
1.2 Background 4
1.2.1 Autonomous Agents 4
1.2.2 Agents in Civil Engineering and Construction 6
1.2.3 Agents in Planning and Scheduling 9
1.2.4 Agents in Simulation 10
1.3 Research Objectives 15
1.4 Expected Contributions 19
1.5 Research Summary 22
1.6 Thesis Organization 23
1.7 Chapter Summary 23

CHAPTER 2: PROTOTYPE AGENT DEVELOPMENT 25

2.1 Introduction 25
2.2 Background and Agent Development 26
2.3 Belief Networks 34
2.3.1 Belief Networks: Definition 36
2.3.2 How Do Belief Networks Function 39
2.3.3 Benefits of Belief Networks in Decision Support for Simulation 43
2.4 Prototype Agent Description 45
2.4.1 The Agent Element 45
2.4.2 The Condition Element 47
2.4.3 The Effect Element 49
2.4.4 The Agent Process 50
2.5 Single Agent Simulation Example 50
2.6 Chapter Summary 57

CHAPTER 3: AGENT DEFINITION AND APPLICATIONS 60

3.1 Introduction 60
3.2 Intelligence and the Agent Structure 61
3.3 Neural Network Agent 61
3.3.1 Feed-Forward Back-Propagation Neural Networks 62
3.3.2 Neural Network Agent 72
3.4 Algorithm Agent 79
3.4.1 Soil Prediction Algorithm 80
3.4.2 Soil Prediction Agent 83
3.5 Rule-Based Agent 86
3.6 Agent Examples 88
3.6.1 Example 1: Tunnel Construction 88
3.6.2 Example 2: Abutment Construction 93
3.6.3 Example 3: Earthmoving Simulation 98

3.7 CHAPTER SUMMARY 101

CHAPTER 4: INTELLIGENT SIMULATION AGENTS FRAMEWORK 103

4.1 INTRODUCTION 103
4.2 AGENT MODELING APPROACH 104
4.2.1 OVERVIEW 105
4.2.2 INVENTIVE PROBLEM SOLVING 107
4.2.3 AGENT FRAMEWORK DEVELOPMENT 112
4.3 CASE STUDY 121
4.3.1 INTRODUCTION 122
4.3.2 LEAN CONSTRUCTION CONCEPTS 123
4.3.3 TUNNEL CONSTRUCTION EXAMPLE 126
4.4 BENEFITS OF AGENT EMBEDDED MODELING 134
4.5 CHAPTER SUMMARY 135

CHAPTER 5: DISCUSSION 137

5.1 RESEARCH SUMMARY 137
5.2 EVALUATION OF RESEARCH OBJECTIVES 139
5.3 LIMITATIONS 142
5.4 SUMMARY OF RESEARCH CONTRIBUTIONS 143
5.5 RECOMMENDATIONS FOR FUTURE DEVELOPMENT 145

REFERENCES 146

APPENDIX 1 – BOREHOLE DATABASE 158

A1.1 BOREHOLE DATABASE INSTRUCTIONS 159
A1.1.1 SCROLLING THE DATABASE 159
A1.1.2 EDITING DATA 159
A1.1.3 ADDING DATA 160
A1.1.4 EXITING THE PROGRAM 162
A1.2 NEST BOREHOLE DATA 164
A1.2.1 BORE HOLE: TH99-1 164
A1.2.2 BORE HOLE: TH00-2 165
A1.2.3 BORE HOLE: TH99-2 165
A1.2.4 BORE HOLE: TH00-4 166
A1.2.5 BORE HOLE: TH99-3 166
A1.2.6 BORE HOLE: TH00-3 167
A1.2.7 BORE HOLE: TH00-1 167
A1.2.8 BORE HOLE: TH1-5 168
A1.2.9 BORE HOLE: TH99-4 168
A1.3 BOREHOLE DATABASE OVERVIEW 169
A1.4 DATABASE PROGRAM CODE 175

APPENDIX 2 – NEURAL NETWORK DATABASE 179

A2.1 NEURAL NETWORK DATABASE INSTRUCTIONS 180
A2.1.1 THE MAIN FORM 180
A2.1.2 THE DATA EDITING FORM 182
A2.2 NEURAL NETWORK DATABASE OVERVIEW 189
A2.3 DATABASE PROGRAM CODE 193
A2.3.1 PROGRAMMING CODE FOR THE MAIN FORM 193
A2.3.2 PROGRAMMING CODE FOR THE DATA EDITING FORM 197

APPENDIX 3 – BELIEF NETWORK INFORMATION 205

A3.1 OVERVIEW 206
A3.2 BELIEF NETWORK CONSTRUCTION INFORMATION 206
A3.2.1 TRAIN BELIEF NETWORK 206
A3.2.2 PLAY BELIEF NETWORK 208
A3.2.3 TRAINNEST BELIEF NETWORK 232
A3.2.4 LEAN BELIEF NETWORK 234

APPENDIX 4 – TUNNEL TRAINS SIMULATION INFORMATION 266

A4.1 MODEL OVERVIEW 267
A4.2 MODEL CONSTRUCTION 268
A4.2.1 ELEMENT 1 PARENT AGENT ELEMENT 270
A4.2.2 ELEMENT 2 CONDITION ELEMENT 271
A4.2.3 ELEMENT 3 CONDITION ELEMENT 272
A4.2.4 ELEMENT 4 CONDITION ELEMENT 274
A4.2.5 ELEMENT 5 EFFECT ELEMENT 275

APPENDIX 5 – TRUCK QUEUING SIMULATION INFORMATION 277

A5.1 MODEL OVERVIEW 278
A5.2 MODEL CONSTRUCTION 279
A5.2.1 ELEMENT 1 AGENT ELEMENT 281
A5.2.2 ELEMENT 2 SERVER UTILIZATION CONDITION ELEMENT 282
A5.2.3 ELEMENT 3 AVERAGE QUEUE LENGTH CONDITION ELEMENT 284
A5.2.4 ELEMENT 4 AVERAGE QUEUE WAIT CONDITION ELEMENT 286
A5.2.5 ELEMENT 5 IMPROVEMENT CONDITION ELEMENT 288
A5.2.6 ELEMENT 6 LAST ACTION CONDITION ELEMENT 290
A5.2.7 ELEMENT 7 CUSTOMER DELAY CONDITION ELEMENT 291
A5.2.8 ELEMENT 8 INCREASE CUSTOMER EFFECT ELEMENT 292
A5.2.9 ELEMENT 9 INCREASE SERVER EFFECT ELEMENT 294
A5.2.10 ELEMENT 10 DECREASE CUSTOMER EFFECT ELEMENT 296
A5.2.11 ELEMENT 11 DECREASE SERVER EFFECT ELEMENT 298

APPENDIX 6 – BRIDGE ABUTMENT CONSTRUCTION SIMULATION INFORMATION 300

A6.1 Model Overview 301
A6.2 Model Construction 302
A6.2.1 Element 1 Agent Element 304
A6.2.2 Element 2 Average Daily Temperature Condition Element 305
A6.2.3 Element 3 Precipitation Condition Element 306
A6.2.4 Element 4 Precipitation Seven Day Sum Condition Element 307
A6.2.5 Element 5 Production Factor Effect Element 308
A6.3 Modified Task Element Programming Code 309

APPENDIX 7 – EARTHMOVING SIMULATION INFORMATION 318

A7.1 Model Overview 319
A7.2 Model Construction 320
A7.2.1 Element 1 Agent Element 326
A7.2.2 Element 5 Average Daily Temperature Condition Element 327
A7.2.3 Element 6 Precipitation Condition Element 328
A7.2.4 Element 7 Precipitation Seven Day Sum Condition Element 329
A7.2.5 Element 8 Production Factor Effect Element 330

APPENDIX 8 – NORTH EDMONTON SANITARY TRUNK TUNNEL SIMULATION INFORMATION 331

A8.1 Model Overview 332
A8.2 Model Construction 333
A8.2.1 Element 1 Belief Network Agent Element 340
A8.2.2 Element 2 Rule Based Agent Element 341
A8.2.3 Element 3 Rule Based Agent Element 342
A8.2.4 Element 4 Algorithm Agent Element 343
A8.2.5 Element 5 Belief Network Agent Element 344
A8.2.6 Element 6 Belief Network Condition Element 345
A8.2.7 Element 7 Belief Network Condition Element 347
A8.2.8 Element 8 Belief Network Condition Element 350
A8.2.9 Element 9 Belief Network Effect Element 351
A8.2.10 Element 10 Belief Network Effect Element 355
A8.2.11 Element 11 Rule Based Condition Element 358
A8.2.12 Element 12 Rule Based Effect Element 360
A8.2.13 Element 13 Rule Based Condition Element 362
A8.2.14 Element 14 Rule Based Effect Element 365
A8.2.15 Element 15 Algorithm Condition Element 366
A8.2.16 Element 16 Algorithm Effect Element 372
A8.2.17 Element 17 Belief Network Condition Element 373
A8.2.18 Element 18 Belief Network Condition Element 375
A8.2.19 Element 19 Belief Network Condition Element 377
A8.2.20 Element 20 Belief Network Effect Element 378

APPENDIX 9 – AGENT MODEL ELEMENT INFORMATION 379

A9.1 BELIEF NETWORK AGENT 380
A9.1.1 PARENT AGENT ELEMENT PROGRAMMING CODE 380
A9.1.2 CONDITION ELEMENT PROGRAMMING CODE 387
A9.1.3 EFFECT ELEMENT PROGRAMMING CODE 391
A9.2 NEURAL NETWORK AGENT 396
A9.2.1 PARENT AGENT ELEMENT PROGRAMMING CODE 396
A9.2.2 CONDITION ELEMENT PROGRAMMING CODE 416
A9.2.3 EFFECT ELEMENT PROGRAMMING CODE 419
A9.2.4 DYNAMIC LIBRARY LINK 422
A9.2.5 OBSERVATION ASSISTANT ELEMENT PROGRAMMING CODE 435
A9.2.6 OBSERVATION ELEMENT PROGRAMMING CODE 445
A9.3 ALGORITHM AGENT 449
A9.3.1 PARENT AGENT ELEMENT PROGRAMMING CODE 449
A9.3.2 CONDITION ELEMENT PROGRAMMING CODE 478
A9.3.3 EFFECT ELEMENT PROGRAMMING CODE 482
A9.4 RULE BASED AGENT 487
A9.4.1 PARENT AGENT ELEMENT PROGRAMMING CODE 487
A9.4.2 CONDITION ELEMENT PROGRAMMING CODE 491
A9.4.3 EFFECT ELEMENT PROGRAMMING CODE 493

LIST OF FIGURES

CHAPTER 1: INTRODUCTION 1

FIGURE 1-1. CURRENT USE OF SIMULATION IN PROJECT INVESTIGATION 16
FIGURE 1-2. PROPOSED FRAMEWORK FOR EMBEDDED AGENT SIMULATION MODELING 19

CHAPTER 2: PROTOTYPE AGENT DEVELOPMENT 25

FIGURE 2-1. INITIAL DEPICTION OF AGENT COMPOSITION 29
FIGURE 2-2. EXTENSION OF AGENT COMPOSITION 31
FIGURE 2-3. THE AGENT STRUCTURE IN SIMPHONY 32
FIGURE 2-4. EXAMPLE NETWORK 35
FIGURE 2-5. EXAMPLE NETWORK WITH CONDITIONAL PROBABILITIES 37
FIGURE 2-6. SINGLY-CONNECTED OR POLYTREE BELIEF NETWORK 41
FIGURE 2-7. MULTIPLY-CONNECTED BELIEF NETWORK 41
FIGURE 2-8. NETWORK WITH A MEGANODE 42
FIGURE 2-9. CUTSET CONDITIONING 42
FIGURE 2-10. AGENT ELEMENT 46
FIGURE 2-11. PROGRAMMING WINDOW FOR LINKING A PARAMETER 47
FIGURE 2-12. CONDITION ELEMENT 48
FIGURE 2-13. EFFECT ELEMENT 50
FIGURE 2-14. EXAMPLE TRUCK AND SHOVEL OPERATION 54
FIGURE 2-15. BELIEF NETWORK USED FOR RESOURCE ALLOCATION IMPROVEMENT PROCESS 55

CHAPTER 3: AGENT DEFINITION AND APPLICATIONS 60

FIGURE. 3-1 ILLUSTRATION OF NEURON (ABOURIZK 1993) 63
FIGURE. 3-2 ILLUSTRATION OF A NEURAL NETWORK 64
FIGURE. 3-3 ILLUSTRATION OF A PERCEPTRON 66
FIGURE 3-4. EXAMPLES OF PERCEPTRON TRANSFER FUNCTIONS 69
FIGURE 3-5. AGENT ELEMENT 74
FIGURE 3-6. CONDITION ELEMENT 75
FIGURE 3-7. EFFECT ELEMENT 76
FIGURE 3-8. OBSERVATION ASSISTANT ELEMENT 79
FIGURE 3-9. OBSERVATION ELEMENT 79
FIGURE 3-10. AGENT ELEMENT 83
FIGURE 3-11. CONDITION ELEMENT 84
FIGURE 3-12. EFFECT ELEMENT 85
FIGURE 3-13. AGENT ELEMENT 87
FIGURE 3-14. CONDITION ELEMENT 87
FIGURE 3-15. EFFECT ELEMENT 87
FIGURE 3-16. TUNNEL CONSTRUCTION SIMULATION SCREEN CAPTURES 90
FIGURE 3-17. TRAIN BELIEF NETWORK 91
FIGURE 3-18. BRIDGE CONSTRUCTION CPM MODEL AND SIMULATION ABSTRACTION 95
FIGURE 3-19. AGENT EMBEDDED IN SPS EARTH MOVING MODEL 100

CHAPTER 4: INTELLIGENT SIMULATION AGENTS FRAMEWORK 103

FIGURE 4-1. MASTER/SLAVE AGENT STRUCTURE 105
FIGURE 4-2. PORTION OF THE CONFLICT MATRIX. 112
FIGURE 4-3. PROJECT MANAGEMENT MODEL (SOURCE AHUJA ET AL. 1994) 115
FIGURE 4-4. CEM CONFLICT MATRIX 116
FIGURE 4-5. CEM CONFLICT MATRIX IN BELIEF NETWORK REPRESENTATION 118
FIGURE 4-6. TRIZ ABSTRACTION FOR MASTER AGENT'S REASONING 119
FIGURE 4-7. EMBEDDED AGENT FRAMEWORK 120
FIGURE 4-8. CEM CONFLICT MATRIX WITH LEAN CONSTRUCTION PRINCIPLES 126
FIGURE 4-9. AGENT EMBEDDED MODEL FOR NEST TUNNEL CONSTRUCTION 128
FIGURE 4-10. AGENT AND MODEL INTERACTION FLOWCHART 129

CHAPTER 5: DISCUSSION 137

REFERENCES 146

APPENDIX 1 – BOREHOLE DATABASE 158

FIGURE A1-1. BOREHOLE DATABASE PROGRAM FORM 163
FIGURE A1-2. BOREHOLE DATABASE PROGRAM FORM DESIGN VIEW 173
FIGURE A1-3. BOREHOLE DATABASE RELATIONAL DIAGRAM 174

APPENDIX 2 – NEURAL NETWORK DATABASE 179

FIGURE A2-1. NEURAL NETWORK DATABASE PROGRAM MAIN FORM 183
FIGURE A2-2. NEURAL NETWORK DATABASE PROGRAM DATA EDITING FORM 183
FIGURE A2-3. NEURAL NETWORK DATABASE PROGRAM MAIN FORM DESIGN VIEW 191
FIGURE A2-4. NEURAL NETWORK DATABASE PROGRAM DATA EDITING FORM DESIGN VIEW 191
FIGURE A2-5. NEURAL NETWORK DATABASE RELATIONAL DIAGRAM 192

APPENDIX 3 – BELIEF NETWORK INFORMATION 205

FIGURE A3-1. TRAIN BELIEF NETWORK 207
FIGURE A3-2. PLAY BELIEF NETWORK 209
FIGURE A3-3. TRAINNET BELIEF NETWORK 233
FIGURE A3-4. LEAN BELIEF NETWORK 235

APPENDIX 4 – TUNNEL TRAINS SIMULATION INFORMATION 266

FIGURE A4-1. PROJECT LEVEL VIEW 268
FIGURE A4-2. PARENT AGENT ELEMENT 6 CHILD WINDOW 269
FIGURE A4-3. PARENT AGENT MODELING ELEMENT PARAMETERS 270
FIGURE A4-4. CONDITION MODELING ELEMENT PARAMETERS 271
FIGURE A4-5. CONDITION MODELING ELEMENT PARAMETERS 272
FIGURE A4-6. CONDITION MODELING ELEMENT PARAMETERS 274

Figure A4-7. Effect Modeling Element Parameters 275

APPENDIX 5 – TRUCK QUEUING SIMULATION INFORMATION 277

Figure A5-1. Cyclone Parent Element Child Window 279
Figure A5-2. Agent Parent Element Child Window 280
Figure A5-3. Agent Modeling Element Parameters 281
Figure A5-4. Server Utilization Condition Modeling Element Parameters 282
Figure A5-5. Average Queue Length Condition Modeling Element Parameters 284
Figure A5-6. Average Queue Wait Condition Modeling Element Parameters 286
Figure A5-7. Improvement Condition Modeling Element Parameters 288
Figure A5-8. Last Action Condition Modeling Element Parameters 290
Figure A5-9. Customer Delay Condition Modeling Element Parameters 291
Figure A5-10. Increase Customer Effect Modeling Element Parameters 292
Figure A5-11. Increase Server Effect Modeling Element Parameters 294
Figure A5-12. Decrease Customer Effect Modeling Element Parameters 296
Figure A5-13. Decrease Server Effect Modeling Element Parameters 298

APPENDIX 6 – BRIDGE ABUTMENT CONSTRUCTION SIMULATION INFORMATION 300

Figure A6-1. Project Level View 302
Figure A6-2. Agent Parent Element Child Window 303
Figure A6-3. Agent Modeling Element Parameters 304
Figure A6-4. Average Daily Temperature Condition Element Parameters 305
Figure A6-5. Precipitation Condition Element Parameters 306
Figure A6-6. Precipitation Seven Day Sum Condition Element Parameters 307
Figure A6-7. Production Factor Element Parameters 308

APPENDIX 7 – EARTHMOVING SIMULATION INFORMATION 318

Figure A7-1. Project Level View 320
Figure A7-2. Activity Level View 321
Figure A7-3. Dump Location 2 Child Element Window 322
Figure A7-4. Dump Location 3 Child Element Window 323
Figure A7-5. Source Location 4 Child Element Window 324
Figure A7-6. Agent Parent Element Child Window 325
Figure A7-7. Agent Modeling Element Parameters 326
Figure A7-8. Average Daily Temperature Condition Modeling Element Parameters 327
Figure A7-9. Precipitation Condition Modeling Element Parameters 328
Figure A7-10. Precipitation Seven Day Sum Condition Modeling Element Parameters 329
Figure A7-11. Production Factor Effect Modeling Element Parameters 330

APPENDIX 8 – NORTH EDMONTON SANITARY TRUNK TUNNEL SIMULATION INFORMATION 331

FIGURE A8-1. PROJECT LEVEL VIEW 333
FIGURE A8-2. MASTER AGENT ELEMENT 1 CHILD WINDOW 334
FIGURE A8-3. SLAVE AGENT ELEMENT 2 CHILD WINDOW 335
FIGURE A8-4. SLAVE AGENT ELEMENT 3 CHILD WINDOW 336
FIGURE A8-5. SLAVE AGENT ELEMENT 4 CHILD WINDOW 337
FIGURE A8-6. SLAVE AGENT ELEMENT 5 CHILD WINDOW 338
FIGURE A8-7. BELIEF NETWORK AGENT MODELING ELEMENT PARAMETERS 340
FIGURE A8-8. RULE BASED AGENT MODELING ELEMENT PARAMETERS 341
FIGURE A8-9. RULE BASED AGENT MODELING ELEMENT PARAMETERS 342
FIGURE A8-10. ALGORITHM AGENT MODELING ELEMENT PARAMETERS 343
FIGURE A8-11. BELIEF NETWORK AGENT MODELING ELEMENT PARAMETERS 344
FIGURE A8-12. BELIEF NETWORK CONDITION MODELING ELEMENT PARAMETERS 345
FIGURE A8-13. BELIEF NETWORK CONDITION MODELING ELEMENT PARAMETERS 347
FIGURE A8-14. BELIEF NETWORK CONDITION MODELING ELEMENT PARAMETERS 350
FIGURE A8-15. BELIEF NETWORK EFFECT MODELING ELEMENT PARAMETERS 351
FIGURE A8-16. BELIEF NETWORK EFFECT MODELING ELEMENT PARAMETERS 355
FIGURE A8-17. RULE BASED CONDITION MODELING ELEMENT PARAMETERS 358
FIGURE A8-18. RULE BASED EFFECT MODELING ELEMENT PARAMETERS 360
FIGURE A8-19. RULE BASED CONDITION MODELING ELEMENT PARAMETERS 362
FIGURE A8-20. RULE BASED EFFECT MODELING ELEMENT PARAMETERS 365
FIGURE A8-21. ALGORITHM CONDITION MODELING ELEMENT PARAMETERS 366
FIGURE A8-22. ALGORITHM EFFECT MODELING ELEMENT PARAMETERS 372
FIGURE A8-23. BELIEF NETWORK CONDITION MODELING ELEMENT PARAMETERS 373
FIGURE A8-24. BELIEF NETWORK CONDITION MODELING ELEMENT PARAMETERS 375
FIGURE A8-25. BELIEF NETWORK CONDITION MODELING ELEMENT PARAMETERS 377
FIGURE A8-26. BELIEF NETWORK EFFECT MODELING ELEMENT PARAMETERS 378

APPENDIX 9 – AGENT MODEL ELEMENT INFORMATION 379

LIST OF TABLES

CHAPTER 1: INTRODUCTION 1

CHAPTER 2: PROTOTYPE AGENT DEVELOPMENT 25

TABLE 2-1. UPPER AND LOWER TOLERANCE LIMITS FOR AVERAGE QUEUE WAIT AND LENGTH 51
TABLE 2-2. STATES FOR PERFORMANCE VARIABLES 56
TABLE 2-3. AGENT OBSERVATIONS 59
TABLE 2-4. MODEL PROPERTIES AT OBSERVATIONS 59

CHAPTER 3: AGENT DEFINITION AND APPLICATIONS 60

TABLE 3-1. RESULTS FOR AGENTS ACTIONS IN TUNNEL CONSTRUCTION SIMULATION MODEL 93
TABLE 3-2. WEATHER DATA OF ABUTMENT CONSTRUCTION WITH WEATHER EFFECTS (WALES 1994) 96
TABLE 3-3. REPRODUCED RESULTS FOR ABUTMENT CONSTRUCTION WITH WEATHER EFFECTS 98
TABLE 3-4. AGENT INFLUENCE ON LOADING TIMES IN THE SPS EARTH MOVING MODEL 101

CHAPTER 4: INTELLIGENT SIMULATION AGENTS FRAMEWORK 103

CHAPTER 5: DISCUSSION 137

REFERENCES 146

APPENDIX 1 – BOREHOLE DATABASE 158

TABLE A1-1. DATA FOR BOREHOLE TH99-1 164
TABLE A1-2. DATA FOR BOREHOLE TH00-2 165
TABLE A1-3. DATA FOR BOREHOLE TH99-2 165
TABLE A1-4. DATA FOR BOREHOLE TH00-4 166
TABLE A1-5. DATA FOR BOREHOLE TH99-3 166
TABLE A1-6. DATA FOR BOREHOLE TH00-3 167
TABLE A1-7. DATA FOR BOREHOLE TH00-1 167
TABLE A1-8. DATA FOR BOREHOLE TH1-5 168
TABLE A1-9. DATA FOR BOREHOLE TH99-4 168

APPENDIX 2 – NEURAL NETWORK DATABASE 179

TABLE A2-1. DATA FOR WEATHER NEURAL NETWORK 184

APPENDIX 3 – BELIEF NETWORK INFORMATION 205

Table A3-1. Belief Network Node Property Descriptions Table 207
Table A3-2. CurrentObsGTLast Node Conditional Probability Table 207
Table A3-3. CurrentObsGT2 Node Conditional Probability Table 207
Table A3-4. CountGT1 Node Conditional Probability Table 208
Table A3-5. AddTrain Node Conditional Probability Table 208
Table A3-6. Belief Network Node Property Descriptions Table 209
Table A3-7. LastAction Node Conditional Probability Table 209
Table A3-8. CD Node Conditional Probability Table 209
Table A3-9. QL Node Conditional Probability Table 210
Table A3-10. QW Node Conditional Probability Table 210
Table A3-11. SU Node Conditional Probability Table 210
Table A3-12. Improvement Node Conditional Probability Table 210
Table A3-13. DecreaseCustomers Node Conditional Probability Table 210
Table A3-14. IncreaseCustomers Node Conditional Probability Table 215
Table A3-15. DecreaseServers Node Conditional Probability Table 220
Table A3-16. IncreaseServers Node Conditional Probability Table 226
Table A3-17. Belief Network Node Property Descriptions Table 233
Table A3-18. CurrentOBSGTLast Node Conditional Probability Table 233
Table A3-19. CurrentOBSGT2 Node Conditional Probability Table 234
Table A3-20. PoorTBMUtlization Node Conditional Probability Table 234
Table A3-21. AddTrain Node Conditional Probability Table 234
Table A3-22. Belief Network Node Property Descriptions Table 235
Table A3-23. Scope Node Conditional Probability Table 235
Table A3-24. Time Node Conditional Probability Table 236
Table A3-25. Cost Node Conditional Probability Table 236
Table A3-26. Quality Node Conditional Probability Table 236
Table A3-27. Communications Node Conditional Probability Table 236
Table A3-28. Resources Node Conditional Probability Table 236
Table A3-29. Safety Node Conditional Probability Table 236
Table A3-30. Risk Node Conditional Probability Table 237
Table A3-31. Priciple1 Node Conditional Probability Table 237
Table A3-32. Priciple2 Node Conditional Probability Table 242
Table A3-33. Priciple3 Node Conditional Probability Table 248
Table A3-34. Priciple4 Node Conditional Probability Table 254
Table A3-35. Priciple5 Node Conditional Probability Table 260

APPENDIX 4 – TUNNEL TRAINS SIMULATION INFORMATION 266

APPENDIX 5 – TRUCK QUEUING SIMULATION INFORMATION 277

APPENDIX 6 – BRIDGE ABUTMENT CONSTRUCTION SIMULATION INFORMATION 300

APPENDIX 7 – EARTHMOVING SIMULATION INFORMATION 318

APPENDIX 8 – NORTH EDMONTON SANITARY TRUNK TUNNEL SIMULATION INFORMATION 331

APPENDIX 9 – AGENT MODEL ELEMENT INFORMATION 379

Chapter 1: Introduction

1.1 Introduction

Construction Engineering and Management (CEM) promotes the evolution of the construction industry from a craft culture to a science based on methods and principles. This evolution is occurring as a result of many innovations generated by CEM research. Whereas originally master builders populated the industry, employing themselves to design, plan, and construct a given facility, the modern industry employs a wide range of specialists. These specialists include general contractors, sub-contractors, architects, designers, and project management teams. Each contributes to the project, working together with the others to move a concept from an identified need to an operational facility. One of the current evolutionary movements in the CEM research community is the development of technology to reduce and manage the variability and inherent randomness found in many of today's construction projects. The research presented herein is situated within this evolutionary movement.

The primary impetus for this research is the further advancement of contemporary construction project methods in estimation, execution, and management, specifically regarding the precision and proficiency of these methods. AbouRizk and Mohammed (2002) note that the most recent advancements in this area took place in the late 1950s with the development of techniques based on bar charts and scheduling networks (CPM, PERT, Precedence Networks, etc.). AbouRizk and Mohammed (2002) also point out that the effectiveness of these methods is limited and that they do not provide the construction industry with the ability to "properly plan, estimate, and execute a project in a consistent, efficient, and reliable manner." Similar observations have been made by the International Group for Lean Construction (1999) and Tommelein (1998). Furthermore, statistical summaries on cost and schedule overruns from Abd Majid and McCaffer (1998) and a study examining estimates performed on 213 City of Edmonton construction projects by AbouRizk (1998) identified that these current practices do not provide acceptable tolerances for project control. The intention of this research is therefore to bring about a new paradigm in project planning, estimation, and execution.

This new paradigm addresses the need identified by AbouRizk and Mohammed (2002) for "an ideal (or perfect) project execution plan". The ideal project execution plan could be viewed as a production plan that is flexible and in need of continuous change and improvement. The current methods for project estimation, planning, and execution, based on bar charts and scheduling networks, are too rigid in their structure and do not compliment the dynamics found in real-life construction. In general, what ultimately results is the schedules and plans created do not reflect what actually occurs during the project life. Thus the agent-based problem solving method proposed in this research will provide a new means through which problems can be solved in a manner that reflects the dynamics of real-life construction. This will be accomplished through the use of embedded agents who will carry out artificial simulation real-time decision-making during the course of the simulation model run(s).

In general, the current state of simulation modeling allows simulation modelers to observe a system, model that system on a computer, artificially observe that system in operation, and make necessary changes to the model in an attempt to achieve an improved or optimized modeled system. Van Tol (2000) and McCabe (1997) developed methods of automating this improvement process by combining simulation modeling environments with external artificial intelligence tools in the form of belief networks. There are also numerous examples, as will be identified in the following sections, where agents have been embedded in simulation models to provide an automated real-time simulation improvement. This is accomplished by focusing on the evaluation of modeling object parameter variables. While these approaches may provide an improved or optimized simulation model at a specific level, they do not necessarily provide an overall improved or optimized representation of the modeled system.

The research aims at developing a method or framework for embedding simulation models with autonomous agents and thereby to provide an outlined and structured approach for researchers to build intelligent agents in simulation. These agents would not only be able to improve simulation models through model object parameter evaluation, but would also make forward progress towards evaluating and acting upon the topology of the model. This framework would then lend itself to future work, including the development of the perfect execution plan, where agents within the models would provide improved representations of the systems being modeled.

1.2 Background

1.2.1 Autonomous Agents

There are many synonyms for autonomous agents, such as "intelligent agents", "software agents", and "agents". Along with this assortment of names, the concept of an agent has a broad spectrum of descriptions, allowing the term "agent" to be used openly and thereby contributing to the progressive ambiguity of its definition. Weiss (1999) notes this consequence and asserts that without some degree of definition the term loses all meaning and becomes similar to terms such as "user-friendly". Therefore, since the agent is a major component of this research, a solid definition of an autonomous intelligent agent shall be adopted. This definition is outlined in the following paragraphs.

According to Turban and Aronson (2001) the concept of an "agent" arose in 1950 with Vannevar Bush's vision of a machine, the so-called "Memex" machine. The Memex was conceptualized as a machine that could aid humans to read and organize huge amounts of data and information using associative links. John McCarthy developed 'Advice Taker' in the 1950s (Turban and Aronson 2001). Advice Taker was a software robot (i.e. agent) that navigated networks of information developing over time. This agent even took the necessary steps, such as asking for advice, when its searches were halted. Some of the more recent prototype agents, such as Apple's Phil and Microsoft's Bob, function similarly to Advice Taker. Since Advice Taker's development in the 1950s, influencing factors, such as the growth in computing power, the increase in data collection capabilities, globalization of markets, development of large networks, and the internet, have led to significant developments and the widespread application of autonomous agents. Numerous agent applications can now be found in a variety of environments, such as in word processors that employ agents to correct spelling as the user types, data management systems that use agents to move, collect, and/or analyze data, and game theory applications in which agents assume roles in models such as competitive auctions and bidding processes.

Franklin and Graesser (1996) reviewed agent definitions in the literature and suggested that the diversity in the definition of the term "agent" derives from the specific set of examples for which each agent has been developed. That is, the definition for each agent manifests itself in the specific role and application of that agent. Franklin and Graesser (1996) presented several definitions as they were found in the literature, and developed their own formalized definition as well, which states:

> "An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future."

This definition was developed to encompass all types of autonomous agents, including computer systems, robotics, any other recognizable agents, and non-standard agents such as the thermostat. Keeping this and the other definitions presented by Franklin and Graesser (1996) in mind, the features common to most agent definitions are as follows:

- The term "agent" itself suggests that the agent is acting or empowered to act on behalf of someone or something else. For example, in this research the agents would be acting on behalf of the simulation modeler.

- The agents are situated in an environment and are able to perceive that environment.
- The agents are self-starting autonomous objects, which means they act independently and have an agenda or set of goals to achieve. They are proactive in their actions.
- The agents are persistent and dedicated to their agenda.
- The agents have some degree of intelligence allowing them to perceive their environment and act on it, and a capacity for reason in order to solve problems, draw inferences, and determine actions.
- There is a social nature to the agents allowing them to interact with their environment, including with other agents, such that information collection/transfer is coordinated.

For the purposes of this research, an agent will be regarded as having all the above qualities and behavior.

1.2.2 Agents in Civil Engineering and Construction

Numerous examples of intelligent agent applications in civil engineering can be found in the available literature. Within the scope of this research, the following section provides insight into the state of contemporary agent development for civil engineering planning and scheduling purposes.

Kim and Russell (2003a, 2003b) developed an autonomous system for earthworks called Intelligent Earthworks System (IES). In a very general sense, the system made use of agents, distributed artificial intelligence, Global Positioning Satellites (GPS), sensor and sensing technology, wireless communications technology, and path planning technology to create an autonomous earthworks equipment fleet capable of planning and executing construction operations with little human involvement.

Tserng and Lin (2003) developed the Electronic Acquisition Model for Project Scheduling (e-AMPS). The focus for the development of e-AMPS was to facilitate the coordination and scheduling of projects that used a high number of sub-contractors. It was identified that, generally, sub-contractors have no contractual relationships between each other. This leaves the general contractor or project manager with the large responsibility of ensuring effective communication and coordination between all parties involved. Therefore, e-AMPS was developed as a message agent, or centralized information agent, focusing on the acquisition of external scheduling data and to provide project participants with scheduling information.

Jones and Riley (1994) and Ganeshan et al. (1995) developed systems that made use of agents during the design phase of a project to aid in design evaluation. In these systems the agents worked with the users to make decisions and to evaluate alternatives. In the process of making these decisions, constructability issues, such as time and cost, were also taken into consideration.

Procura (Goldmann 1996) is a project management program developed to support project design planning and scheduling. It uses agents to maintain the design plan and schedule as the design progresses. The program consists of application agents able to make design decisions regarding the project schedule and capable of maintaining records of decision rationales. This allows for design updating and project rescheduling.

Khedro et al. (1995) outlined a research effort put forth by the Construction Engineering Research Laboratories of the US Army Corps of Engineers (USACERL) and four major universities (Carnegie Mellon University, Massachusetts Institute of Technology, Stanford University, and the University of Illinois at Urbana-Champaign) focused on developing infrastructure for facility design and construction. One project of note being carried out by this group was the Agent Collaboration Environment (ACE). The BLAST/ACE model uses agents to analyze thermal behavior of buildings and provide information for improvement. The ACE/Owner/Project Manager model makes use of agents to establish the initial functional requirements for a building, to place the proposed building on a site conceptually, and to pass the information on to an architectural design system.

Pitt, Anderton, and Cunningham (1996) used agents to define a new project management paradigm to support projects in a collaborative, decentralized, and inter-organizational manner. Their efforts produced the GOAL Cooperative Services Framework that made use of agents to distribute messages regarding the intentions and requirements of users (project members) and applications (project software) during their participation in project activities and tasks.

Khedro (1999) developed the CIFE World environment, which is described as an environment for the collaborative distributed building and construction planning of office buildings. CIFE World employed a federated agent construct encompassing a variety of software applications used in design through construction. These applications were integrated into one overall system and viewed as agents. Communication between these applications, or agents, was facilitated through system programs called facilitators.

1.2.3 Agents in Planning and Scheduling

There are many applications in the manufacturing industry in which agents have been used in real-time planning and scheduling for shop floor production. Some systems have used a bidding-based approach, such as those described by Teredesai and Ramesh (1998), Shi-ping et al. (2002), and Brennan and O (2000). In these systems the agents collaborated and competed against each other in attempts to find the most favorable shop floor production system. Akbulut and Sagar (2000) addressed shop floor scheduling with an Agent-Based Scheduling System (ABSS) that employed a global performance monitoring strategy. This approach achieved affirmative shop floor scheduling by using a performance evaluation metric and providing a cooperative environment for the agents that supervised the manufacturing machines to work in. Other approaches taken for improved shop floor scheduling are suggested by Huang et al. (2001), Gu et al. (1995), and Biswas and Merechawi (2000). The systems developed by these researchers focused on using the agents for quick, automated responses to changes in the shop environment. Creighton and Nahavandi (2002) then developed a reinforcement-learning agent that interacted with a discrete-event simulation model to determine optimal operating policies in a multi-part serial line. The reinforcement-learning method was a simulation approach in which the agent learned over multiple simulation runs. The agent would receive information about its environment and select an action that would impact the environment's state by changing the model parameters. The agent would then learn from this action based on the overall state and running cost of the simulation run. This process was carried out for several runs and the agent progressively learned the consequences of its actions with each new run. Ultimately, the agent learned enough from prior runs to produce an improved model.

1.2.4 Agents in Simulation

Two basic agent application branches in simulation can be found in the literature. One branch of agent-based simulation regards agents as a means of managing distributed simulation systems. Wilson et al. (2000) and Szymanski and Chen (2000) have developed agent-based distributed simulation modeling frameworks that make use of agents to coordinate and communicate the various individual simulation model schedules and model information with other models.

The second identified branch of agent-based simulation looks at the use of agents as simulation modeling components providing further definition to the simulation models. These agents are intelligent objects situated within a simulation model capable of observing the model, reacting to changes in the model, and acting upon the model. Agent-based simulation applications in this sense are widespread. Sanchez and Lucas (2002) have identified agent-based simulation efforts for biological modeling, sociological modeling, industrial applications, and military applications. Since this area is closely related to the focus of this research proposal, the following sections describe the state of the art in the various areas of agent-based simulation modeling.

1.2.4.1 Aviation

The MITRE Corporation Center for Advanced Aviation Systems Development (CAASD) is involved in simulation modeling of airspace for air traffic control at the strategic (long-term) and tactical (short-term) levels (Schaefer et al. 2002). As a response to strategic modeling, they have developed the intelligent agent-based model for policy assessment of collaborative traffic flow management (IMPACT). This model addresses probabilistic traffic flow management and uses agents in the model for a decision analysis framework. These agents represent the individual airlines, the FAA, and their method of operation with regard to uncertain weather forecasts and conditions. A spin-off from the IMPACT model was the Jet:Wise model. This model makes use of agents to define airline characteristics, such as fares, schedules, and aircraft size, based on heuristic rules to deal with items such as flight delay, flight cost, and passenger demand.

Lee, Pritchett, and Goldsman (2001) posited the modeling of the US national airspace as a collection of discrete event simulation models and continuous time models. Each individual model, such as the flight of a plane, had impacts on other models, such as ground crews or air traffic controllers. To coordinate the model interactions, an agent architecture was implemented where the agents communicated with each other and directed their simulation models based on the information received from other simulation models.

1.2.4.2 Military

Simulation modeling, virtual reality, and intelligent agents have found a solid niche as military applications. These tools allow modelers to observe, hypothesize, learn, train, strategize, and artificially explore the high-risk area of warfare.

Whelan et al. (2002) identified a cost saving application for agents by including them in a virtual reality system developed for large-scale training mission simulations. The simulation method is directed towards the virtual training of military personnel. In this application, the agents replaced real, physical people, who were previously required to aid in the reality of the training simulation, adding further definition to the simulation.

Another area of agent-based simulation that is of considerable interest to the military is war game simulation. Cares (2002) highlighted the significance of agent use within war game simulation models by noting that agents have the ability to learn progressively over multiple simulation runs, to develop behaviors based on learning these iterations, to allow for improved war gaming through these developed behaviors, and ultimately to provide improved evaluation of possible strategies. In addition to this, agent development has furthered the definition of war game simulation models. King et al. (2002) developed a 'Capture the Flag' game, modeled using an Abstract Force Simulator (AFS), to account for coalition attacks that have political and psychological effects focused on fatiguing an enemy. The agents in this model were called "blobs" and represented soldiers or groups of soldiers. They were able to make strategic decisions and, based on the developments made, they were responsive to the political and psychological effects caused by fatigue and recovery. Bullock et al. (2000) developed the Hierarchical Interactive Theatre Model (HITM), a simulation modeling tool that focused on the effects of air strikes during war. In this model the individual components, such as planes, vehicles, and soldiers, were agents possessing the ability to work together in a total effort to achieve goals. Mason and Moffat (2001) developed "command agents" for command and control (C2) decision modeling in their simulation modeling program OACIS (Object Architecture for C2 In Simulation). Lastly, Rasmussen and Chandler (2002) developed MultiUAV, a simulation model in which agents represented operators controlling unmanned aerospace vehicles.

1.2.4.3 Auctions / Bidding Scenarios

The modeling of competitive auctions or bidding scenarios is another area that has benefited from the application of agents. Generally speaking, the agents in these models represent participants within the process and are able to act out their individual approaches or competitive strategies.

Mizuta and Steiglitz (2000) developed a simulation model for online auction bidding behaviors that bridged the gap between theory and experimental work with human subjects. They employed two different types of agents to model the two different types of bidding approaches. The first agent modeled an early bidder who bids at any time during the bidding process. The second agent was described as a sniper and would represent those bidders who bid at the closing seconds of the bidding period in an attempt to win the bid.

Bower and Bunn (2000) developed a simulation model that compared pool-based daily auctions against continuous bilateral trading as applied to competitive electricity pools or exchanges. The two key aspects studied were the uniform versus discriminatory auction issue and the daily versus hourly segmentation of the bidding process. In modeling their markets, they began at a micro level in which autonomous agents represented the individual market of each electricity-generating firm in the model. These agents worked to develop their own bidding strategies, an understanding of their power plants, and an understanding of their market demands.

Xiong, Hashiyama, and Okuma (2002) developed a simulation model that further explored the simulation of electricity exchange markets by creating simulation agents to represent the bidding strategies of suppliers. These agents focused on maximizing a supplier's day-to-day profit as well as satisfying the supplier's target daily utilization rate.

1.2.4.4 Other Models

Agents have also been used to provide further definition to models for economic applications (Dawid et al. 2001, Mizuta and Yamagata 2001), to investigate purchasing activities and policies within an organizational environment (Ebben et al. 2002), to model transportation systems (Schaefer 2001), for process oriented collaborative inventory management (Fu et al. 2000), and to model people competing for jobs in the labor market (Tassier and Menczer 2001).

1.3 Research Objectives

Project management is defined as the "art and science of directing and coordinating human and material resources to achieve stated objectives within the constraints of scope, time, cost, quality, and to the satisfaction of all parties involved" (Ahuja et al. 1994). This definition lies at the core of construction management efforts to effectively deliver a project from conception to completion. In doing this, managers, or management teams, may address several project attributes, such as cash flows, project schedules, project control measures, project operations, and so on, to ensure that the "stated objectives" are achieved.

Simulation modeling has become a reputable tool in construction for exploring these project attributes by providing a means to quantify the randomness inherit in construction and to competently conceptualize and represent the constructs of an issue by way of a model. It allows modelers to artificially explore modeled systems so that these systems can be better understood, optimized, and/or forecasted. This ability gives project managers insight as to how variability and randomness may impact their projects and how it can be managed.

Simulation is generally applied to a construction project as outlined in Figure 1. A project is recognized and then issues of concern are identified. These issues can come about at any time during the project life cycle, and if they are of a complex, repetitive, stochastic, or dynamic nature then they are relevant to simulation modeling. When modeling these issues, the problem at hand, or system, must be observed so that an abstraction can be made and represented through a simulation model. Once the system is modeled, the modeler can then observe and modify it to achieve a better understanding of the system and possibly achieve an improved system in reality.

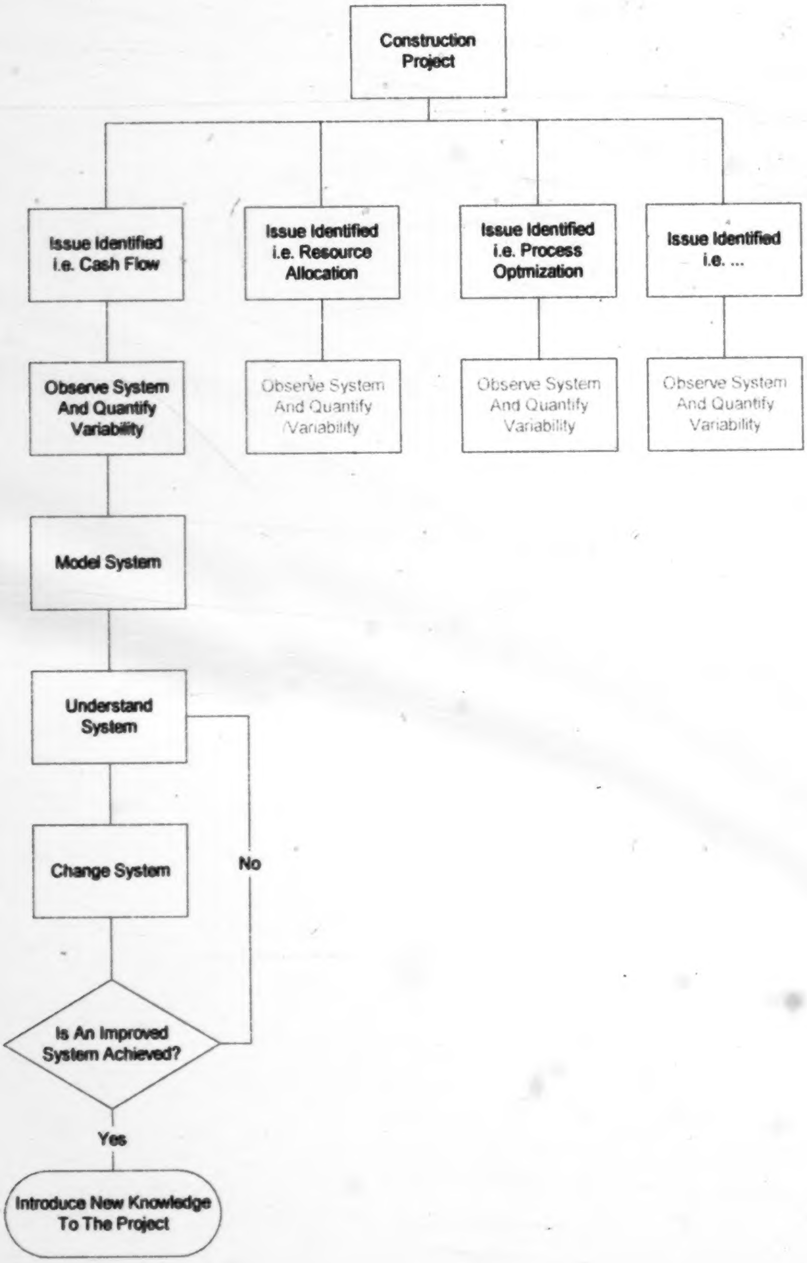

Construction Project
Issue Identified i.e. Cash Flow
Issue Identified i.e. Resource Allocation
Issue Identified i.e. Process Optmization
Issue Identified i.e. ...
Observe System And Quantify Variability
Observe System And Quantify Variability
Observe System And Quantify Variability
Observe System And Quantify Variability
Model System
Understand System
Change System
No
Is An Improved System Achieved?
Yes
Introduce New Knowledge To The Project

Figure 1-1. Current Use Of Simulation In Project Investigation

The current process of understanding the model and improving upon it is labor-intensive for the modeler. The modeler must construct the model, observe the model, find areas of improvement or change for the model, and then repeat the process to the modeler's satisfaction. While this process may not be overly complex, as simulation modeling is becoming easier and easier with object oriented modeling environments, this process does require a fair degree of knowledge on the part of the modeler. To effectively carry out this process, the modeler must be proficient in modeling as well as knowledgeable in the domain of the system being modeled.

Ultimately what is desired is a better representation of the modeled system regardless of the modeler's intentions, whether it is to develop a better understanding of how the system functions or to optimize the system. For example, a proficient modeler could be given the task of modeling a tunnel construction project. While the modeler is capable of modeling the project, the modeler may not know the construction alternatives available to the project. Is the tunnel best constructed using a TBM or should it be hand dug? When should a second train be introduced for optimal spoil removal? When should a night shift be implemented to achieve a timely project completion while minimizing impacts on the surrounding residential community? These questions may need to be addressed by the modeler to better represent the system being modeled. A similar issue can arise for an experienced construction manager who is knowledgeable of the alternatives, but may not know how to effectively model them. Intelligent agents offer the ability to reduce these requirements of the modeler through an agent's ability to provide construction and simulation modeling expertise while assuming a role within the simulation environment to carry out the iterative model evaluation process for the modeler.

The objective of this research is to create a framework for embedding agents in simulation models for construction management issues as outlined in Figure 2. This framework would provide future researchers and modelers with an agent development structure for agent-embedded simulation modeling. This structure would define agent properties, characteristics, abilities, and performances as they would pertain to construction simulation models. In addition to this objective, there are three further sub-objectives that can also be identified. The first of these sub-objectives is the development of a mechanism for the agent so that it can be applied to many different models without complications. That is, the same agent structure must be embeddable in a tunnel construction model as well as in a crushed aggregate production plant model. Following this, the second objective would address agent knowledge encapsulation, allowing the agents to efficiently function in a variety of modeled environments. From the literature review it has been found that there are two basic types of embedded agents used in simulation modeling: intelligent objects that make decisions or take actions within a model to add further definition to the model, and intelligent observers that change parameters within a model for model improvement. The third sub-objective of this research is to develop a third type of embedded agent that is able to observe and act upon the model in terms of its evaluation of object element parameters and the topology of the model.

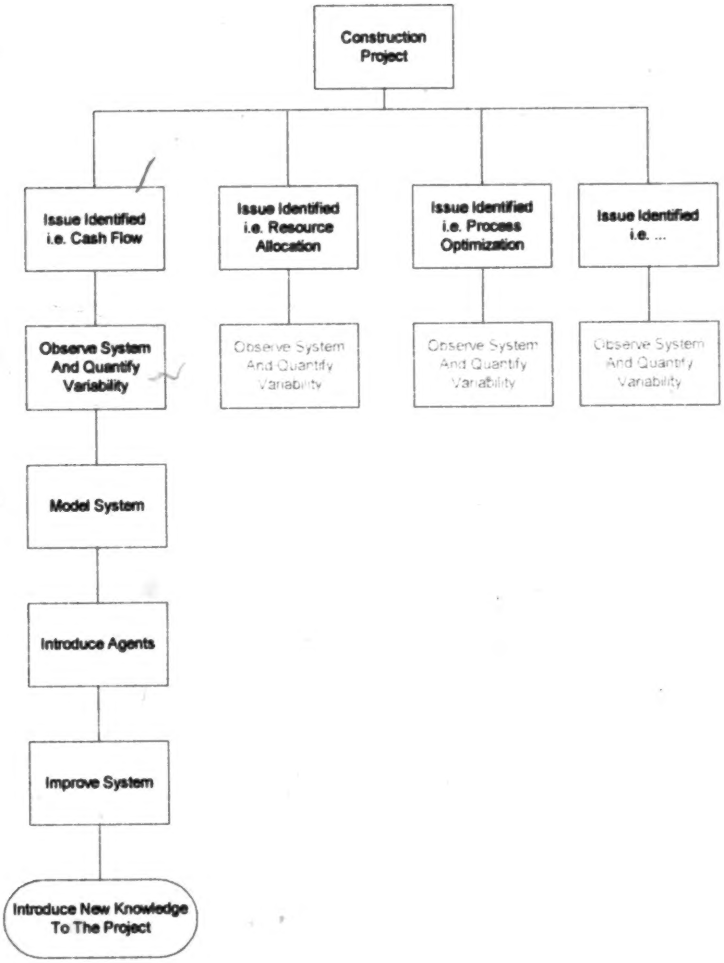

Construction Project
Issue Identified i.e. Cash Flow
Issue Identified i.e. Resource Allocation
Issue Identified i.e. Process Optimization
Issue Identified i.e. ...
Observe System And Quantify Variability
Observe System And Quantify Variability
Observe System And Quantify Variability
Observe System And Quantify Variability
Model System
Introduce Agents
Improve System
Introduce New Knowledge To The Project

Figure 1-2. Proposed Framework For Embedded Agent Simulation Modeling

1.4 Expected Contributions

The previously stated contribution of this research is to develop a framework for agent development in agent-embedded simulation modeling for construction management issues. In doing so, one of the tangible contributions of this research is the development of intelligent simulation models capable of autonomously modifying themselves in order to achieve an overall improved system function and modeled representation. One could compare the intelligent agents in a simulation model to inspectors, project engineers, or superintendents on a construction site. These people observe a system and make real-time changes to that system in order to achieve a better performance. The expected contribution from this agent development framework is a development structure for agents, which can act as "artificial superintendents" or observers, for simulation models by addressing how these models are being created with regard to the objects within the models and the models themselves.

The agent approach described in this research differs from previous simulation model evaluation research efforts, such as that done by Van Tol (2000) and McCabe (1997), by creating a dynamic evaluation tool that is able to react to the changes in the state and condition of the simulation modeling environment. While efforts from individuals such as Van Tol (2000) and McCabe (1997) have produced automated evaluation tools, the programs they have developed are relatively static when compared to the capabilities of the agent framework. By nature, these programs simply take input information from a simulation run and processed it into output information for the next simulation run. The agents developed through this research are designed to function in a more dynamic sense by reacting to conditions in the model during simulation run-time and by trying to influence the model as the simulation progresses.

This agent approach described in this research also differs from the other agent approaches presented in the literature review through the following features. With respect to agent development for civil engineering applications, the agents developed under this research have been done with a focus on providing decision support for construction related issues. For the most part, the agents described in section 1.2.2 are either focused on design evaluation or information management/distribution. With respect to the agents used for real-time planning and scheduling mainly in the manufacturing industry, the agents developed under this research have been created to carry out a similar role, but in a simulation environment. This ability allows for the agents to proactively assume estimating, planning, and execution roles in a virtual environment. This means that the agents are carrying out their roles before any real action is taken. Thus, the information gained from the agent actions can be used to improve what will happen in the future rather than making use of the agents to react to what is going to happen in the immediate future. With respect to the previous research efforts concerning agents in simulation, the intent of this research is to develop a generic agent framework that can be employed to fill a variety of roles. As stated previously, from the information provided in the available literature there are two branches of agent development in simulation. The first branch makes use of agents as managers of distributed simulation modeling environments. The second branch makes use of agents for role playing within a model thereby adding further realism to the model. The intent of this research is to create an agent with these capabilities, but more so, a generic tool within simulation that a modeler can assign any role to through instructions to an agent. Ultimately, this intent provides for an agent that can assume the role of the modeler during the simulation evaluation process. Also by developing an agent structure that is able to make use of a variety of artificial intelligence tools, the agent can improve the decision making capabilities of the modeler by providing access to information the modeler may not possess or by being able to process quantities of information beyond the modelers capabilities.

1.5 Research Summary

In achieving the goal of developing an embedded agent framework for a simulation modeling environment there are two key components to be undertaken. The first component is the development of the agent. The second component is the development of the framework in which the agent will function. The resulting development process was conducted in three distinct phases.

The first phase focused on the development of a prototype agent. The first step in this phase addressed the definition and scope of the agent. This led to the development of a functional structure the agent would use in a modeling environment and the eventual development of the prototype agent. The resulting prototype agent was able to perform satisfactorily within a modeling environment using a belief network facility for its intelligence and reasoning.

The second phase of development embellished the previous work to provide agents able to use artificial neural networks, algorithms, and rule based intelligence faculties. Effort was then made to investigate the agent's mobility between modeling environments as well as the agent's ability to transpose itself between modeling environments with similar issues.

The last phase of development focused on the agent framework. This phase initially required the investigation of inventive problem solving theory (TRIZ) so that TRIZ concepts could be applied to the framework. The investigation was followed with the development of the embedded agent framework. The resulting framework allowed numerous agents to be employed in a modeling environment and focus on models with a micro- and macro-level outlook. The framework allowed each individual agent to investigate and act on small (micro-level) portions of a model while communicating and coordinating with each other so that a global (macro-level) perspective of the model was maintained.

1.6 Thesis Organization

The organization of the thesis closely follows that of the research development phases. Chapter 2 discusses the definition of the agent, its structure, and how it was developed for use in a simulation modeling environment. Chapter 3 explores the embellishments made to the prototype agent that allow it to use other forms of intelligence and function in any modeling environment. Chapter 4 details the framework developed for the embedded agent modeling process. Chapter 5 contains the final discussion that concludes this thesis.

1.7 Chapter Summary

The main undertaking of this research is to develop a new problem solving pattern that will provide improved proficiency and precision when solving construction engineering and management issues. This new method in simulation modeling has been identified as agent embedded simulation modeling. Agent embedded simulation modeling makes use of simulation modeling to attack precision concerns while leaving the proficiency matters to agent theory.

Simulation modeling has proven itself a beneficial tool for the construction industry through its continual validation as a fit for the conceptualization of construction issues by means of aptly modeling construction systems with their randomness and variability. While simulation can provide managers with improved confidence in how they manage construction issues, a problem that has been brought to light is the ease, or proficiency, with which people can use simulation. For construction managers, their main skill set is their expertise in construction. Currently, simulation modeling is not considered a conventional tool in this area of expertise. For the simulation modeler, modeling is their mainstay and construction experience is often limited. Clearly both groups would benefit from the use of an intelligent assistant that could aid them when using simulation modeling to solve problems. Thus it is the intention of the agent framework to create an intelligent modeling environment that can aid a person, regardless of their expertise, in problem solving when using simulation.

The end result is the creation of an intelligent modeling environment where the models are able to address themselves. The use of simulation modeling provides improved precision in relation to contemporary construction problem solving methods and the agent's ability to reason and perform intelligently preserves a method to ensure proficiency when modeling.

Chapter 2: Prototype Agent Development

2.1 Introduction

The objective for this research is to develop a framework for embedded agent simulation modeling. The opening assignment in completing this task is to develop the primary building block for this framework, the agent. To flesh out the agent, several considerations must be made. One of these considerations is to determine what characteristics and abilities characterize an agent. Thought must be also given to how the agent is to function in relation to the scope of the research topic and problem. These two steps will conceptualize a definition of the proposed agent for this research and provide perspective on how the agent should be represented and function within the simulation modeling environment. This fundamental perception of the agent will provide a foundation from which a prototype agent can be developed. This chapter summarizes this process and the steps involved in the creation of a functional prototype agent.

The prototype agent described in this chapter makes use of a belief network intelligence facility for its reasoning. The choice of this intelligence for the prototype agent was based purely on the researcher's familiarity with belief networks. Other approaches to the agents intelligence, such as rule bases, could be taken to solve this problem in a similar fashion. The agent structure developed allows for exchangeable intelligence faculties, which will be elaborated upon in the following chapter.

This chapter is organized as follows: section 2.2 provides insight into the requirements established for the agent development and how these requirements are met in the prototype agent structure developed. Section 2.3 provides information regarding belief networks. Section 2.4 describes the prototype agent developed along with an explanation of the modeling elements associated with the agent and the mechanics of these elements. Section 2.5 provides a validating example of the prototype agent where the agent is employed in a simulation modeling environment to assess resource allocation concerns in an earthmoving activity. Section 2.6 summarizes the chapter.

2.2 Background and Agent Development

Section 1.2.1 of the previous chapter presented a formalized definition for an agent as well as a list of agent features commonly found in the literature. From this information an agent could be generally characterized as a social and intelligent object that is able to perceive its environment and act upon it based on its agenda, which is assigned to the agent by its employer. The definition and common features found in section 1.2.1 provide a technical definition for the agent that can translate into specifications for the agent development. Ultimately, the agent must possess the properties and characteristics outlined in section 1.2.1 to be considered an agent.

A second consideration to add to the agent development specification is the need for a method for embedding intelligent agents in simulation modeling environments to conduct real-time simulation model evaluations. Conceptually, the agents must be "intelligent observers" or "artificial construction superintendents" that can be placed within a modeling environment where they can inspect and make real-time decisions regarding a modeled system's performance. In addition to this, the benefits from the agents implemented for the modeled system must be able to extend back to the original real-life system. This expectation begins to further define the agent's role in the simulation modeling environment. The agent's influence must be able to extend back to real-life, which means the agents may be required to perceive an environment outside the simulation modeling environment. For instance, an agent may be required to assess real-life production data found outside of the simulation modeling environment so that the agent can determine actions to be carried out in the simulation environment. Therefore, in terms of the agent perceiving its environment, the agent environment is now defined as the modeling environment, the modeler employing the agent, and any information sources that could possibly be found outside of the modeling environment.

In terms of the agent's actions, the two key actions demanded of the agent are the ability to communicate and the ability to act upon the model. The agent must be able to communicate with other agents, the modeler, and external sources so that the beneficial information obtained from the agent is taken advantage of and shared. In terms of acting on the model, the agent must be able to influence all aspects of the model. This means that the agent must have the ability to role play within the model, thereby adding further definition to the model, evaluate and modify modeling element parameters, and act on the topology of the model. This last ability means that the agent must be able to add, remove, and modify modeling elements, entities, and relationships. Lastly, the agent must be able to move from one modeling environment to the next as well as from problem to problem with little effort by the modeler. The concept of this framework would diminish significantly if the agents had to be custom developed for every individual application. Therefore an agent with mobility, implying that the agent is able to easily move from one environment to the next, and transposability, implying that the agent's intelligence is easily applied to a variety of similar problems, is desired.

These two previous considerations have now provided a firm idea of what is required for an agent developed under this research. The first consideration established what is required of the agent by its definition. The second consideration established what is required of the agent by this research. The next step is to take the defined perception of the agent and resolve how it will be represented.

The agent is perceived as consisting of three main components. The agent must have a perception facility where it takes in information. The agent must also have an action facility where it is able to carry out tasks such as communicating information, acting upon a model, requesting information, and so forth. The third component ties the perception and action facilities together with an intelligence and reasoning facility. The resulting concept, depicted in Figure 2-1, makes use of these three components to create a conceptual agent structure where the agent is able to perceive something, think about it, and act accordingly.

Figure 2-1 also gives light to the composition of the agent's agenda. The agent's agenda contains the information which the agent uses to identify its goal or objectives. Relating the agenda back to the purpose of this research, the agenda can be considered the instructions given to the agent by the modeler. Figure 2-1 indicates that the modeler would define the agent's agenda by instructing them on what to observe and how to act based on their observations. The intelligence mechanism would not constitute a part of the agent's agenda; it would be a function of the agent that processes the information from the observations so that the agent can determine the appropriate action(s) to be taken.

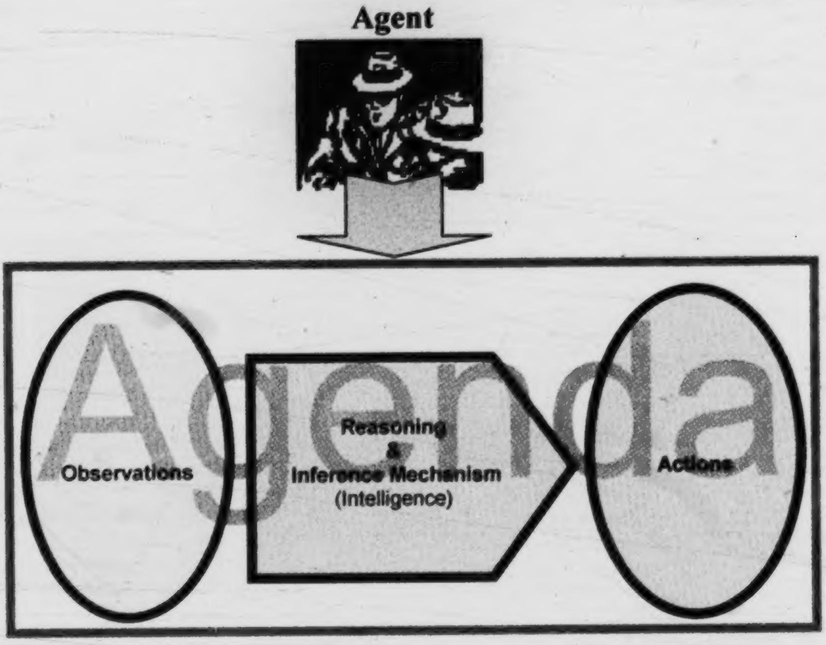


Figure 2-1. Initial depiction of Agent Composition

Extending this initial depiction of the agent composition, the agent can be pictured as an object consisting of two parts. The first part is the functioning portion of the agent. This part contains the essential aspects common to every agent that would allow the agents to function and operate. The second part would be the agent's agenda. This part of the agent is where the modeler instructs the agent and assigns an agenda. Figure 2-2 represents the agent in this manner. The top portion of Figure 2-2 represents the core portion of the agent containing all the workings that allow it to function. Two key components of the agent depicted in Figure 2-2 are the agent's intelligence facilities and the agent's observation schedule. The agent's observation schedule instructs the agent how often or under what circumstances the agent should make observations of its environment. The agent's intelligence facilities allow the agent to process the information from its observations and determine as to what action should be taken. The two components should be presented to the modeler so that the modeler can determine the type of intelligence to be used by the agent and influence when and how the agent makes its observations.

The modeler should also have influence over how the agent functions in the modeling environment as to what information must be found by the agent, how the agent makes its observations, how this information should be presented to the agent's intelligence faculties, what actions can be carried out by the agent, and under what circumstances the actions should be carried out. These properties can be defined through the agent's agenda. The bottom portion of Figure 2-2 better defines what will frame the agent's agenda. The agenda will consist of a set of conditions and a set of effects. It is through these sets of conditions and effects that the modeler will instruct the agent as to what are its purposes and goals. The set of conditions will be items that the agent should be observing set by the modeler. For example, in a model where the main concern is production, the set of conditions would include items such as system throughput, equipment utilization, idle time, and so forth. The set of effects would outline to the agent what possible actions the agent could carry out. This is where the modeler could instruct the agent that, if it has inferred from its observation that a server in the system is causing poor production because it does not have enough clients to keep it constantly active, the agent could add more clients to the system.

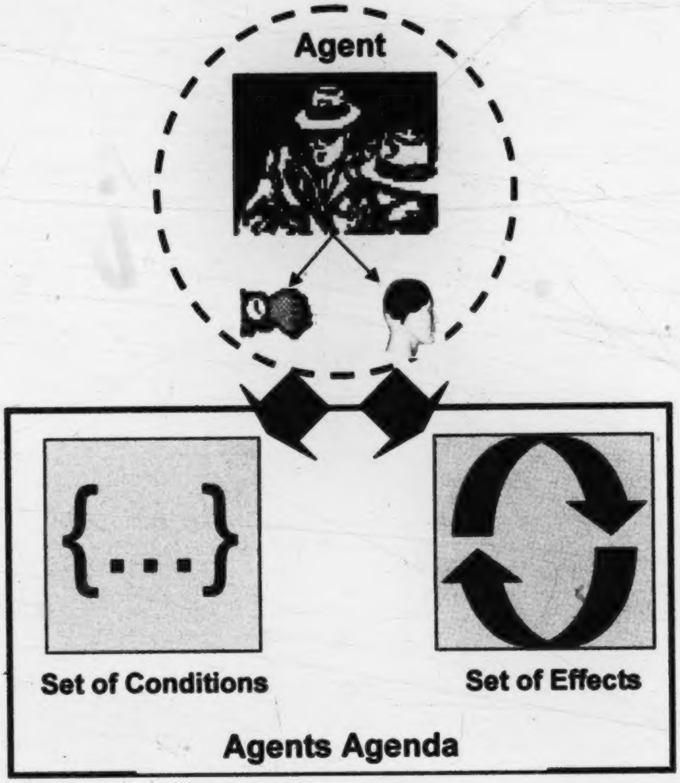


Figure 2-2. Extension of Agent Composition

The agent concept has now evolved into an intelligent nucleus with a 'notebook' where the modeler provides the agent a list of all the items the agent should be observing and a list of all the possible actions that the could be carried out based on its observations.

The agent representation in Figure 2-2 provides a structure that can be developed within a simulation modeling environment. The *Simphony* modeling environment (Hajjar and AbouRizk 2002) allows for hierarchical modeling, which means that a modeling element placed within the modeling environment can be composed of other modeling elements in a parent/child modeling structure. Figure 2-3 shows this hierarchical modeling structure and how the depiction of the agent in Figure 2-2 can be translated into a modeling object. Figure 2-3 shows the agent in the modeling environment as a parent element for a collection of child elements that represent a set of conditions and effects. This structure allows a modeler to embed the agent in the modeling environment and then define the agent's agenda through condition and effect modeling elements in the agent's child window.

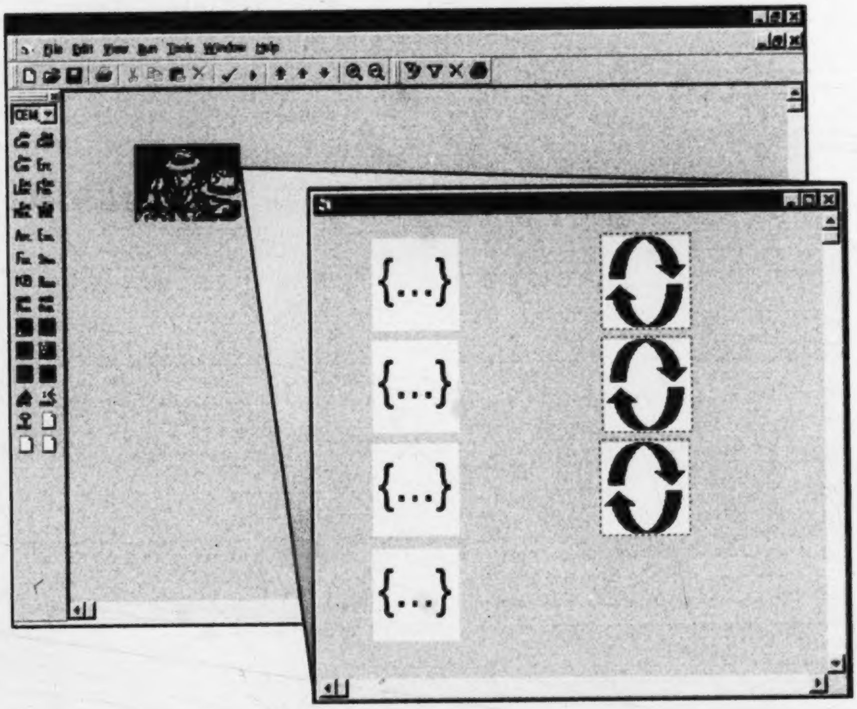


Figure 2-3. The Agent Structure in Simphony

The structure depicted in Figure 2-3 was adopted for the prototype agent's development, with the parent agent element as the engine for the entire structure. This modeling element administrates all the agent's actions. The *condition* child elements contain information regarding what observations are to be made in the model as well as information related to what the observations mean to the agent. The *effect* child elements outline the possible actions that the agent could carry out as well as information related to the circumstances required for the actions to be carried out.

The structure presented in Figure 2-3 is fairly generic, as it was conceptualized with this intent. The structure allows for the agent's intelligence to be changed in two ways. First, the structure allows the agent's intelligence to change with regard to the type of intelligence used. In one circumstance an agent may be best served by a neural network as its intelligence faculty. In another circumstance the agent may be better served by a belief network intelligence. Second the structure allows for the modeler to assign a specific domain of intelligence for the agent. Using a neural network example, it is known that one neural network can not be constructed to address every possible problem; neural networks are constructed for specific problems. A neural network designed for determining concrete productivity is different from a neural network constructed for determining weather conditions. The structure developed allows for this determination of domain knowledge within the chosen agents intelligence faculty.

Attention must be also given in the development of the agent's intelligence as to whether the domain is too specific. An agent would not be beneficial if the neural network developed for its intelligence was only applicable to one specific simulation model. An agent would be most effective if it makes use of artificial intelligence faculties, such as neural networks, that are constructed in a manner such that it can be applied in a general sense. For example, one could construct a neural network designed for manpower resource scheduling rather than a neural network designed for pipefitter resource scheduling at refinery plant X for company Y during a shutdown in 2001. This practice will be further addressed in the following chapter. It should also be noted that designing the intelligence in general terms would also aid in transposing agents from one modeling environment to the next with little modification to the agenda by the modeler. This will also be detailed in the next chapter.

The developed agent was first created using a belief network as its intelligence faculty. This was to develop an agent that was able to immediately function in a modeling environment and that, with further development, could be made to allow for other types of artificial intelligence to be used. To aid the reader in understanding the developed prototype, the following section will provide an introduction to belief networks.

2.3 Belief Networks

A belief network can be described as a network diagram consisting of a series of variables that describe a domain of interest (See Figure 2-4). The nodes in the network represent domain variables. These variables are defined by their states. The states vary according to the variables being described. For example a node called "Hot" could have the states of "Yes" and "No", whereas a node called "Temperature" could have the states of "Cold", "Warm", and "Hot". The arcs in the network diagram represent the relationships between the variables.

The function of a belief network is to determine the probability, or belief, of a proposition given various bits of information about the network. This information is used to define the states of the nodes in the network. The proposition is the question asking the probability of a specific node being a given state. When determining a proposition, the belief network makes use of prior and posterior probabilities (Russell and Norvig 1994). Prior probabilities are determined before any network information is given. A posterior, or conditional, probability is determined after information has given insight to the proposition.

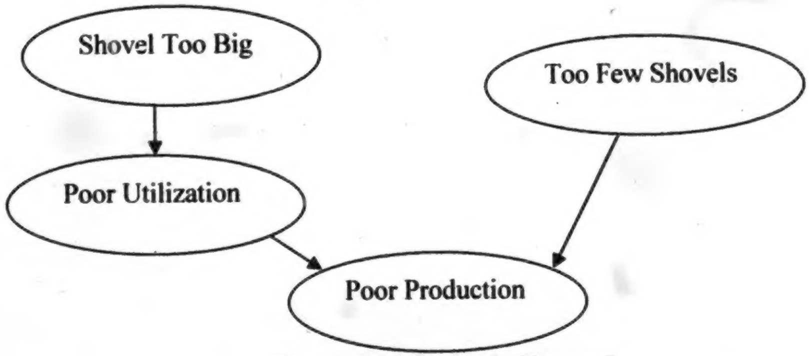


Figure 2-4. Example Network

The difference between prior and posterior probabilities can be best explained using an example: the probability of an ace of spades being drawn from a deck of cards. The prior probability for this problem is 1 card out of 52, or 2%. If it were known that the drawn card was a spade, the posterior probability would become 1 card out of 13, or 8%. The difference between the posterior probability and the prior probability is that the posterior probability is dependent upon the states of variables while a prior probability is not. This is important since it allows the belief network to function without knowing the states of all the variables in the network.

2.3.1 Belief Networks: Definition

A belief network can be defined as a graph in which the following hold (Russell and Norvig 1994):

- A set of variables describing the domain of interest make up the nodes in the network.
- A set of distinct links or arcs connect pairs of nodes. These arcs have one direction, from parent node to child. These arcs reflect the relationships between nodes. An arrow going from node A to node B indicates that node A has a direct influence on node B. Nodes that are not directly connected are independent of each other, also known as d-separation.
- Each node has a conditional probability table that quantifies the effects the parent nodes have on that node. Thus, for each state of the parent nodes (e.g. true or false) the resulting state of the child node must be defined. If a child has more than one parent, this must be done for each combination of the parent node states.
- The graph is a directed, acyclic graph (DAG); thus, there are no cycles to be found in the graph. The relationships must flow from the parent nodes to their children, as it does not allow for the states of the child nodes to influence the states of their parents.

A belief network can also be described as a representation of the joint probability distribution (Russell and Norvig 1994). The joint probability distribution specifies the probability assignments for all propositions in the domain (i.e. every possible combination or query for the various states of the variables in the network). The belief network provides a 'snapshot' of the joint probability distribution for specific cases defined by the states of the variables in the belief network. The advantage the belief network provides over the joint probability distribution is a reduced effort for network construction. If a network were to consist of variables that had two possible states, either true or false, the number of calculations required to define the joint would be 2^n, where "n" is the number of variables in the domain (Russell and Norvig 1994). This indicates that as the number of the variables increases, the number of calculations grow exponentially for the joint probability distribution. The belief network, on the other hand, simply requires the definition of a conditional probability table for each node, representing geometric growth.

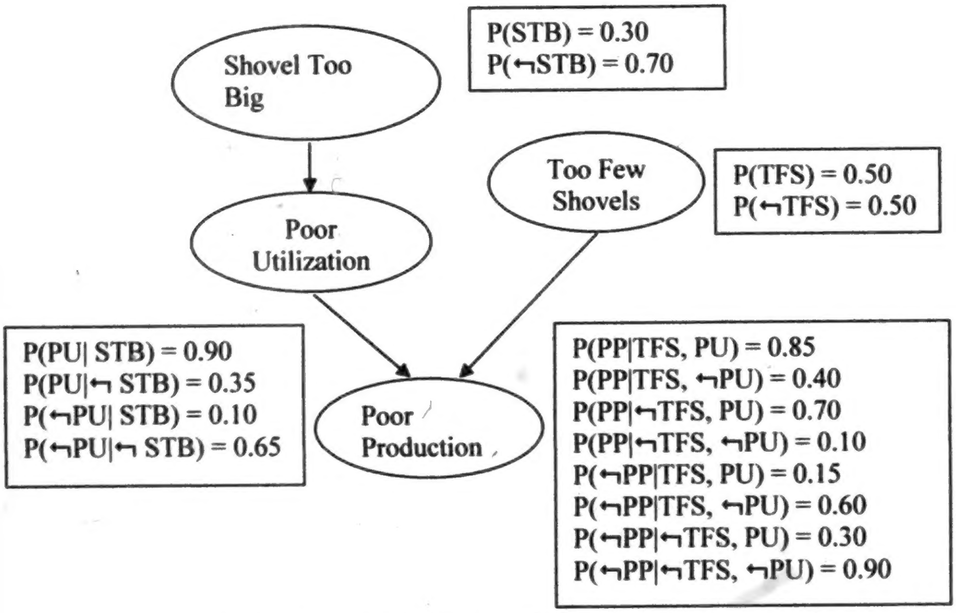

Shovel Too Big
P(STB) = 0.30
P(¬STB) = 0.70
Too Few Shovels
P(TFS) = 0.50
P(¬TFS) = 0.50
Poor Utilization
P(PU| STB) = 0.90
P(PU|¬ STB) = 0.35
P(¬PU| STB) = 0.10
P(¬PU|¬ STB) = 0.65
Poor Production
P(PP|TFS, PU) = 0.85
P(PP|TFS, ¬PU) = 0.40
P(PP|¬TFS, PU) = 0.70
P(PP|¬TFS, ¬PU) = 0.10
P(¬PP|TFS, PU) = 0.15
P(¬PP|TFS, ¬PU) = 0.60
P(¬PP|¬TFS, PU) = 0.30
P(¬PP|¬TFS, ¬PU) = 0.90

Figure 2-5. Example Network with Conditional Probabilities

Figure 2-5 is an example of a network with conditional probability tables defined for the nodes. The notation is as follows: P(¬PP|TFS, ¬PU) = 0.60 states that the probability of 'Poor Production' (PP) being false is 0.60, given that 'Too Few Shovels' (TFS) is true and 'Poor Utilization' (PU) is false. '¬' indicates a variable being false. Similarly P(STB) = 0.30

states that the probability of 'Shovel Too Big' (STB) being true is 0.30 given no further information. If 'Shovel Too Big' was known to be true during an evaluation, this value could be changed to a more appropriate probability. For example, if there were a 90% belief that 'Shovel Too Big' was true then this value could be changed to 0.90. Probabilities can only range between 1 (full belief) and 0 (no belief), and all the probabilities for the states must add up to one. In the case of 'Poor Production' in Figure 2-5, the probabilities for 'Poor Production' being true and false given 'Too Few Shovels' and 'Poor Utilization' are true are 0.85 and 0.15 respectively, which add up to 1. This is simply due to the fact that if there is a 0.85 belief in a proposition being true, then there is a 0.15 belief in that proposition being false. It should also be reinforced that a belief is not a truth. A 0.85 belief in 'Poor Production' being true suggests an 85% belief that this is true and does not suggest 'Poor Production' is 85% true.

Previous research has defined methods for constructing belief networks (Poole et al. 1998; Russell and Norvig 1994). The methods ensure that the above-mentioned properties of a belief network are maintained. The process is as follows:

- Define all the relevant variables that describe the domain.
- Choose an ordering for the variables. Root variables, the variables that have no parents, are ordered first. These are followed by the variables that have the root variables as parents. Nodes that have the previously listed nodes as parents are listed next and so on until all the variables are listed.

- As each variable is removed from the list and placed in the network, arcs should be drawn from the previously placed nodes to the recently placed node. Once this is done, the conditional probability table for the node should be defined.

2.3.2 How Do Belief Networks Function

Belief networks make use of Bayes' rule for probabilistic interference. Bayes' rule is used to calculate conditional probabilities. It allows one to calculate probabilities of events given the probabilities of other events. Bayes' rule is given in Equation 2.1 (Russell and Norvig 1994). Equation 2.1 determines the probability of B given information about A. This equation, however, does require the prior probability of both A and B and the posterior probability of A given B.

$$P(B \mid A) = \frac{P(A|B)\,P(B)}{P(A)} \qquad (2.1)$$

Where:

P(A) = Prior probability of A

P(A|B) = Probability of A given B

P(B) = Prior probability of B

P(B|A) = Probability of B given A

Equation 2.2 is an expanded form of Bayes' theorem able to handle multiple influences (McCabe 1997).

$$P(B_i A) = \frac{P(A \mid B_i)P(B_i)}{\sum_{k=1}^{n} P(A \mid B_k)(PB_k)} \qquad (2.2)$$

Where:

$P(B_i|A)$ = Probability of current node B given A

$P(A|B_i)$ = Probability of A given current node B

$P(B_i)$ = Prior probability of current node B

i = Current node

k = Succeeding nodes

n = Number of succeeding nodes

$P(A|B_k)$ = Probability of A given succeeding node B

$P(B_k)$ = Probability of succeeding node B

Belief networks can be used for essentially four types of inferences (Russell and Norvig 1994). *Diagnostic inferences* look at effects to causes; e.g. what is the probability of a car starting (cause) given the battery is assumed to be dead (effect)? *Causal inferences* look at causes to effects; e.g. what is the probability of the battery being dead (effect) given the car does not start (cause)? *Intercausal inferences* look between causes of a common effect; e.g. what is the difference in probability of the car starting (cause) given that the battery is assumed dead (effect) against the probability of the car starting (cause) given that the battery is assumed dead (effect) and that the radio works (additional cause)? Lastly, a *mixed inference* is any combination of the above.

When evaluating a belief network, the first step is to assess the network. Belief networks can be classified into singly-connected networks (or Polytrees) and multiply-connected networks (Figure 2-6 and Figure 2-7) (Russell and Norvig 1994). The difference between these two networks is in their construction. A singly-connected network has only one path between any two nodes. A multiply-connected network has some nodes that are connected by two or more paths, as in Figure 2-7.

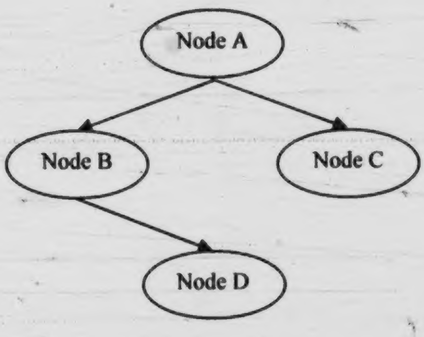


Figure 2-6. Singly-connected or PolyTree belief network

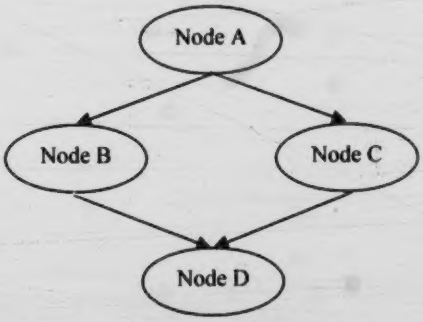


Figure 2-7. Multiply-connected belief network

Singly-connected networks allow for exact solutions to be produced with the application of Bayes' rule through the network (Russell and Norvig 1994). Multiply-connected networks, however, are solved with approximation methods such as clustering, conditioning, and stochastic simulation (Russell and Norvig 1994). Clustering transforms the multiply-connected network into a singly-connected network by combining multiple nodes into a single mega-node (Figure 2-8). Cutset conditioning looks at transforming the multiply-connected network into several simpler singly-connected networks (Figure 2-9), then sets the variables to specific states and calculates the network through weighted averages over the various singly-connected networks. Stochastic simulation makes use of simulation to randomly pick variable states in the domain and determine the proposition solutions from the frequency of occurrences observed during the simulation.

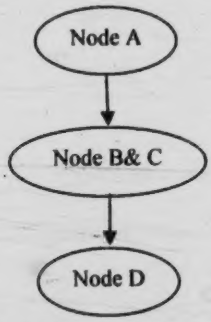


Figure 2-8. Network with a Meganode

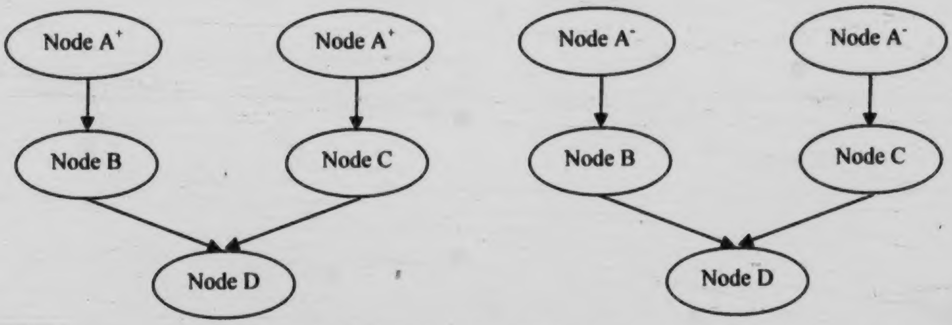


Figure 2-9. Cutset Conditioning

2.3.3 Benefits of Belief Networks in Decision Support for Simulation

As with every artificial intelligence tool, belief networks have an ideal area of application that compliments its talents. McCabe (1997) has identified several properties of belief networks that distinguish them against other forms of artificial intelligence: increased flexibility and ease of network construction, ease in network modification, and flexibility in handling and evaluating data.

In terms of network design and construction, belief networks differ from neural networks or rule-based expert systems by their flexibility in handling input data. Belief networks can be constructed for a wide range of input datasets that can be updated at any time during an evaluation. By comparison, neural networks have a rigid structure designed for specific datasets and rule-based expert systems have a rigid structure that requires data entry at specific locations during an evaluation. Belief networks also offer flexibility in terms of ease of construction over rule-based expert systems with their cause-to-effect perspective. This is the opposite of rule-based expert systems, which are constructed effect-to-cause. This process is often found to be more intuitive for those developing the network and therefore requires less development effort. Lastly, another feature of belief networks that set them apart from other forms of artificial intelligence is that they make use of expert opinion for their construction and, unlike neural networks, they do not require historical data.

Belief networks are also favorable with respect to implementing modifications to the network. The addition or deletion of variables to a belief network is accomplished with little impact to the rest of the network. The variables are simply added or deleted and the affected conditional probability tables are adjusted. In the case of rule-based expert systems, the rule base must be examined and adjusted. In the case of neural networks, new data sets must be acquired and the network must be reconstructed, retrained, and retested.

Information flow in a belief network is very flexible. The belief network can accept inputs and provide outputs at any point during an evaluation. Since belief networks make use of prior and posterior probabilities, they are able to function with incomplete or missing data. By contrast, rule-based expert systems require information at specific points in their diagnosis in order to provide the desired output. Neural networks have a rigid input structure that requires a specific input data set.

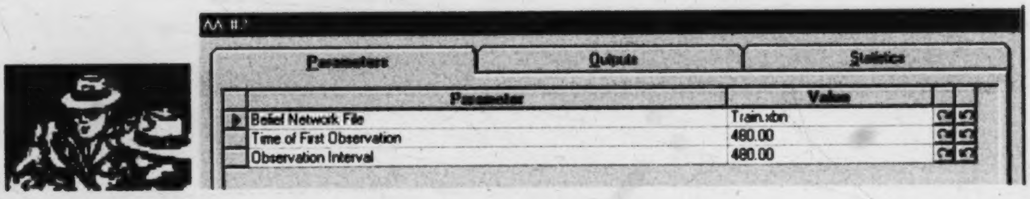
In terms of this research, the main benefit that belief networks provide for simulation modeling is a flexible intelligence facility for the agents. The universal goal of the agent is to provide a service that will relieve the simulation modeler of the effort and knowledge required to observe, analyze, and modify a simulation model. The belief network is able to provide this service through its ability to take any data set relevant to its domain, however incomplete or limited, and make inferences. For example, a large all-encompassing belief network could be constructed to describe the domain of equipment maintenance. This network could then be applied to models involving mining truck operations. Since the belief network does not require a structured input vector or input process, this same network could be then applied to models concerned with the maintenance of rock crushers. The maintenance of rock crushers would be considerably different from that of mining trucks. This would not pose a large problem for the agents as they would focus on another portion of the belief network that better suited the issues faced in rock crusher maintenance. Thus the data handling properties of the belief network allow for the intelligence captured in the network to be applied to many different modeling environments with minimal complications.

2.4 Prototype Agent Description

The development structure of the prototype agent consists of three modeling elements, with the agent element being the parent element for the condition and effect child elements, as shown in Figure 2-3. The course of action for this structure during a model observation is to have the parent agent conduct the observation process by first looking to its condition child elements to determine what observations are required and then using its intelligence to process this information before looking to its effect child elements to determine what actions it should take. This process will be discussed in detail in this section, along with a description of the modeling elements developed for the agent structure. The modeling elements were developed in the simulation modeling environment *Simphony* (Version 1.05) with a Dynamic Link Library (DLL) taken from Microsoft® Research's Bayesian Network Authoring and Evaluation Tool MSBNx (1.4).

2.4.1 The Agent Element

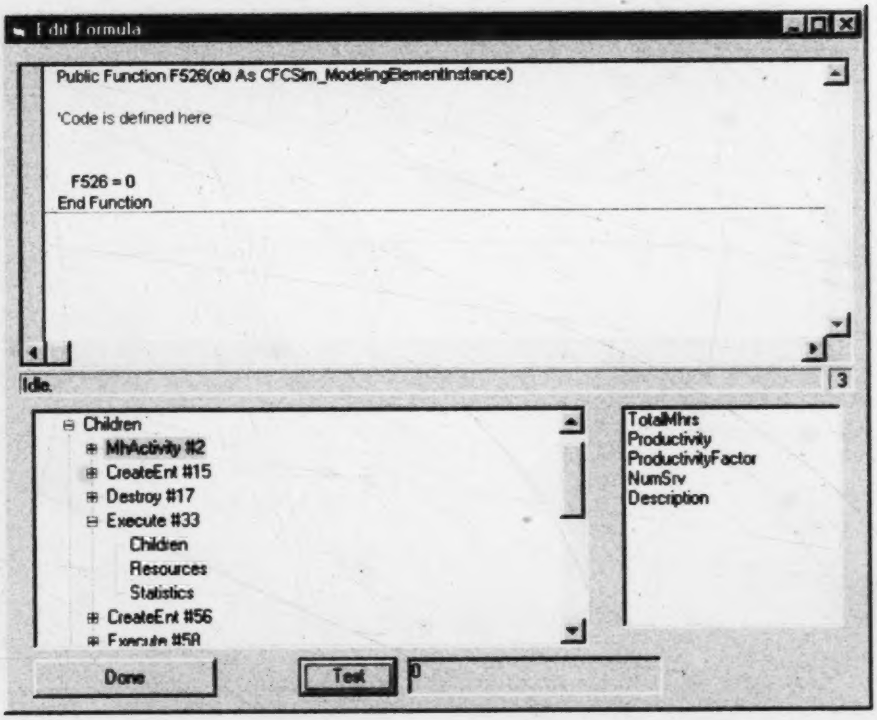
The prototype agent element is presented in Figure 2-10. The element has three parameters required from the modeler. The first parameter is the "Belief Network File". This parameter is where the modeler selects the belief network that the agent is to use for its intelligence. The parameter is defined when the modeler selects a belief network file from a drop down list box associated with the parameter. The list box refers to a data folder located on the hard drive of the modeler's computer. The data folder's location on the computer is where the collection of the various belief network files are stored. The next parameter the modeler is required to define is the "Time of First Observation". This parameter, as its name suggests, is where the modeler instructs the agent as to when it should initiate its first observation. The modeler defines this parameter by simply typing in a number to represent the number of simulation time units that must progress before the first observation is to occur. The last parameter required for the modeling element is the "Observation Interval". This parameter allows the modeler to instruct to the agent what time unit interval is required between agent observations occurring after the first observation. The modeler defines this parameter by entering a number.



Parameter	Value
Belief Network File	Train.xbn
Time of First Observation	480.00
Observation Interval	480.00

Figure 2-10. Agent Element

The last two parameters establish the observation schedule for the agent. If the modeler were to define these parameters with numbers, as shown in Figure 2-10, the observation schedule would be very rigid: the first observation would occur at 480 simulated time units and the subsequent observations would occur at 960, 1440, 1920, and so forth. This rigid structure is not the only option for the agent's observation schedule, as these parameters can be linked. A parameter is linked when its definition refers to user-defined pseudo-Visual Basic program code or formulae. Linking a parameter is accomplished by clicking on the arrow button to the far right of the parameter. In Figure 2-10 it can be seen that there are two arrow buttons on the far right of parameter. The button on the left is used to link the parameter. The button on the right is used to remove the link. When a parameter is linked, the modeler is presented with a programming window as shown in Figure 2-11. Here is where the modeler can create dynamic observation schedules based on states of the simulation model or any formulae the modeler can envision.



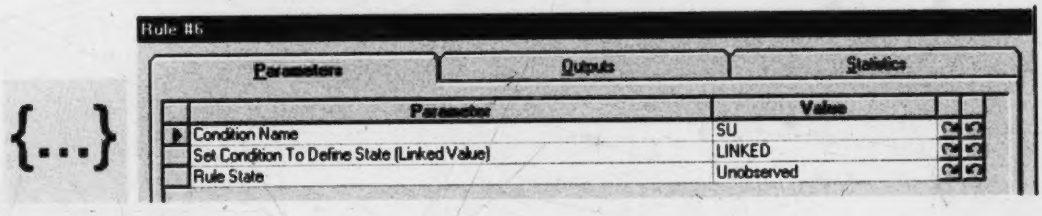
```
Public Function F526(ob As CFCSim_ModelingElementInstance)

'Code is defined here


  F526 = 0
End Function
```

Edit Formula
Idle.
3
Children
MhActivity #2
CreateEnt #15
Destroy #17
Execute #33
Children
Resources
Statistics
CreateEnt #56
Execute #58
TotalMhrs
Productivity
ProductivityFactor
NumSrv
Description
Done
Test
0

Figure 2-11. Programming Window for Linking a Parameter

2.4.2 The Condition Element

The condition element and its three parameters are presented in Figure 2-12. The first parameter is the "Condition Name". This parameter references the condition element to a specific node within the belief network. The modeler defines this parameter by selecting the node from a drop down list of nodes found in the belief network. The next parameter is the "Set Condition to Define State (Linked Value)". This parameter is where the modeler identifies what observations are to be made and how these observations will determine the previously identified nodes state in the belief network. This parameter must be defined as a linked parameter and will provide the definition of the last parameter, "Rule State". The modeler is to make a linked reference in the "Set Condition to Define State (Linked Value)" parameter through programming code as to the observations that need to be made and how the observations will determine the state of the belief node defined in the "Rule State" parameter. As an example of this concept, the modeler could define the link for the "Set Condition to Define State (Linked Value)" parameter as follows: if the simulation clock is less than 500 time units then the "Rule State" parameter will be defined as 'early'. If the simulation clock is equal to or over 500 time units then the "Rule State" parameter will be defined as 'late'. If an observation cannot be made, then the "Rule State" parameter will be defined as 'unobserved'. Carrying out this process defines the "Rule State" parameter during simulation runtime.

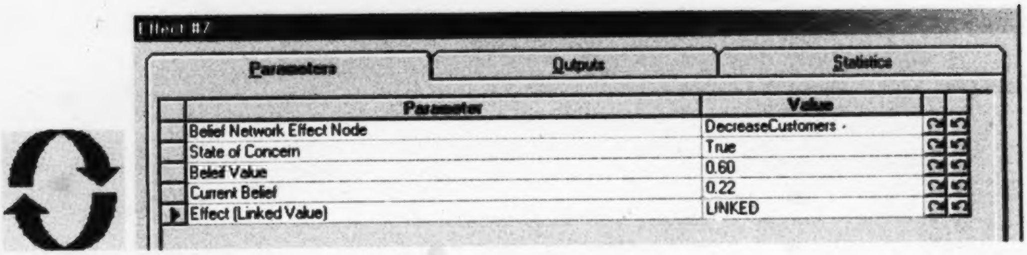


Parameter	Value
Condition Name	SU
Set Condition To Define State (Linked Value)	LINKED
Rule State	Unobserved

Figure 2-12. Condition Element

2.4.3 The Effect Element

The effect element (Figure 2-13) contains the action instructions for the agent and has five parameters. The first parameter is the "Belief Network Effect Node". This parameter references the effect element to a node in the belief network, selected from belief network nodes in a drop down list. The next parameter is the "State of Concern". This parameter is used to identify the state of the previously-defined belief node where the agent should consider carrying out the action found in the effect element. This parameter is defined by selecting the belief node state from a drop down list that contains the states for the prescribed node. The next parameter is the "Belief Value". This parameter defines a tolerance for the belief value of the state of the concerned node. It is this tolerance value that must be exceeded during an evaluation of the belief network for the agent to consider carrying out the action found within the effect element. Figure 2-13 shows a condition element in which the belief in the node "DecreaseCustomers" being true must exceed 0.60 for the agent to consider carrying out the action found in this condition element. The parameter following the "Belief Value" parameter is the "Current Belief" parameter. This parameter simply displays the belief of the specified nodes state during simulation runtime. It is of no concern to the modeler. The last parameter is the "Effect (Linked Value)" parameter. This is where the modeler defines what actions are to be carried out if the agent has found the effect element to be relevant. This parameter is a linked parameter which allows the modeler great flexibility in the actions that can be carried out by the agent.



	Parameter	Value
	Belief Network Effect Node	DecreaseCustomers
	State of Concern	True
	Belief Value	0.60
	Current Belief	0.22
▶	Effect (Linked Value)	LINKED

Figure 2-13. Effect Element

2.4.4 The Agent Process

The mechanics that allow the agent to function follow a specific process. The agent identifies that an observation should take place. The parent agent element looks through its child elements and finds its condition elements. The parent agent instructs the condition elements to carry out their observations and determine how these observations will affect the nodes found in the parent agent's belief network. This information is then passed from the condition elements to the parent agent element. The parent agent inputs this information into the belief network and conducts an evaluation of the network. Once the evaluation is carried out, the parent agent looks again to its child elements to find the effect elements. The agent uses the information from the belief network evaluation to determine if any effect elements should be considered. If an effect element is found to be valid, the parent agent instructs that element to carry out its effect or action.

2.5 Single Agent Simulation Example

The validation example presented is a simple queuing problem abstracted from McCabe (1997). Further details regarding this example can be found in Appendix 5.

The example is a truck and shovel earthmoving operation. The operation is allowed to make use of one shovel and a fleet of trucks that can vary in number between two and ten. The problem for the simulation model is to determine the appropriate number of trucks for the shovel based on the queuing occurring within the model. The average arrival rate for the trucks is 4.5 trucks per hour, and the average service rate for the trucks is 31.1 trucks per hour. Both of these activities were modeled using exponential distributions. The cost ratio of shovel to truck is 2 to 1. The upper and lower limits for the server utilization are 0.25 and 0.95 respectively. The upper limit for the customer delay index is set at 0.3. The limits for the Average Queue Length and Average Queue Wait are dependent on the number of customers in the system and are given in Table 2-1.

Table 2-1. Upper and Lower Tolerance Limits for Average Queue Wait and Length

Number of Customers	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Lower Wait Limit	0.019	0.064	0.139	0.251	0.410	0.618	0.882	1.201	1.569
Upper Wait Limit	0.141	0.432	0.857	1.389	2.019	2.725	3.497	4.315	5.160
Lower Length Limit	0.094	0.307	0.664	1.180	1.880	2.762	3.822	5.038	6.370
Upper Length Limit	0.243	0.735	1.440	2.311	3.333	4.471	5.709	7.019	8.367

The example model used here is evaluated based on queuing properties in the model and the model's performance. This queuing evaluation makes use of four performance indices (McCabe 1997). These indices are calculated at each queue node in the model. They are the

Average Queue Wait (eq. 2.3), Average Queue Length (eq. 2.4), Server Utilization (eq. 2.5), and Customer Delay Index (eq. 2.6). They are defined as follows:

$$QW = \frac{\sum w}{n} \tag{2.3}$$

$$QL = \frac{\sum [l_n * T_n]}{S} \tag{2.4}$$

$$SU = \frac{P}{(S_f - S_s)} \tag{2.5}$$

$$CD = \frac{\sum D}{C} \tag{2.6}$$

Where:

C = Cycle time

CD = Customer delay index

D = Delay length experienced by a customer during its cycle

l_n = Queue length n

n = Number of entities to come to a queue

P = Server productive time

QL = Average queue length

QW = Average queue wait

S = Simulation time

S_F = Server finish time

S_S = Server start time

SU = Server utilization

T_n = Time spent at queue length queue length n

w = Wait experienced by an entity in a queue

The resulting simulation model is given in Figure 2-14. The problem was modeled with a modified CYCLONE simulation modeling template developed by Van Tol (2000). The model illustrates an earthmoving activity with a loading task, where queuing took place, and a truck haul, or truck cycle, task. The validation of the simulation model was done for a simulation run using four trucks. The server utilization was found to be 0.48, while Carmicheal (1987) had determined a utilization of 0.49, resulting in an error of 2.04%. The backcycle time is determined by subtracting the service time from the cycle time. The cycle time was found to be 0.26 and the service time was found to be 0.03. Therefore the simulated backcycle time was calculated at 0.23. Carmicheal (1987) specified a backcycle time of 0.2222 hours (4.5 trucks per hour) and a service time of 0.032 hours (31.1 trucks per hour). The degree of error for the backcycle and service times are -3.51% and 6.25% respectively. Thus, with such small errors, the simulation model appeared to be simulating adequately.

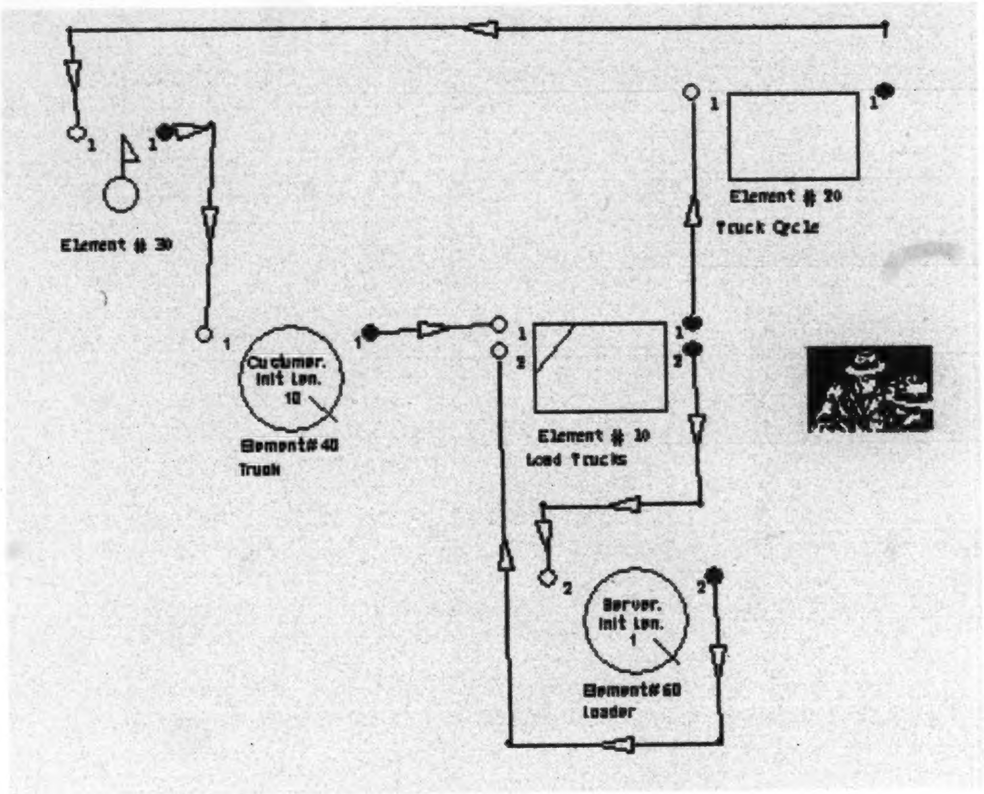

Element # 30
Element # 20
Truck Cycle
Customer.
Init Len.
10
Element#40
Truck
Element # 10
Load Trucks
Server.
Init Len.
1
Element#60
Loader

Figure 2-14. Example Truck and Shovel Operation

The belief network design is shown in Figure 2-15. The states for nodes in the network are presented in Table 2-2. Essentially, the belief network was built with six nodes dedicated to model observations and four nodes for model recommendations. The observation nodes include the ‘CD’, ’SU’, ’QW’, ’QL’, ’Improvement’, and ’LastAction’ nodes. The ‘CD,’ ’SU,’ ’QW,’ and ’QL’ nodes were used to describe queuing properties of the simulation model for the Customer Delay Index, Server Utilization Index, Average Queue Wait Index, and Average Queue Length Index respectively. The ’Improvement’ and ’LastAction’ nodes observed the last actions upon the simulation model and indicated whether these actions had a positive or negative influence. The recommendation nodes include the ‘IncreaseServers,’ 'DecreaseServers,' 'IncreaseCustomers,' and 'DecreaseCustomers' nodes. They act upon the model as their name suggests, with the trucks being customers and the shovel being the server, as it would be described in queuing theory.

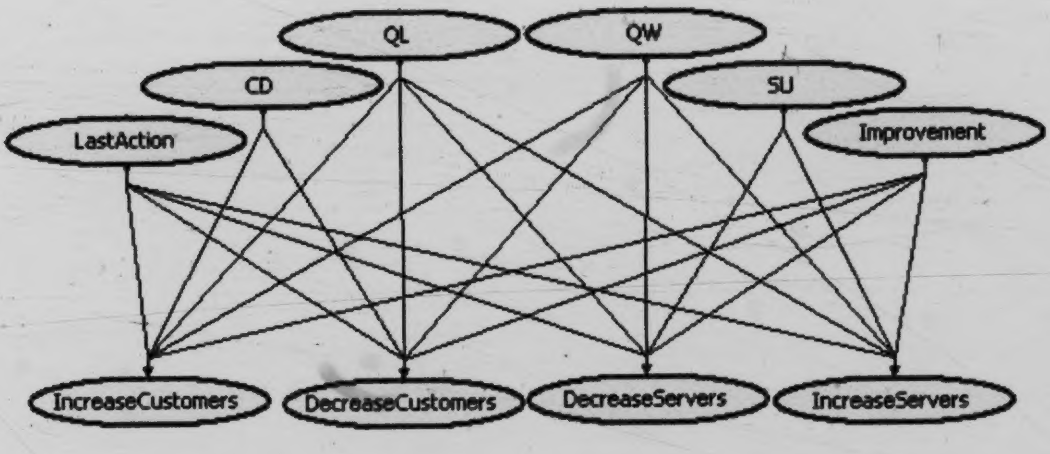

QL
QW
CD
SU
LastAction
Improvement
IncreaseCustomers
DecreaseCustomers
DecreaseServers
IncreaseServers

Figure 2-15. Belief Network Used for Resource Allocation Improvement Process

Table 2-2. States for Performance Variables

Customer Delay (CD)	$CD < CD_U$
	$CD > CD_U$
Server Utilization (SU)	$SU_L < SU < SU_U$
	$SU < SU_L$
	$SU > SU_U$
Queue Wait (QW)	$QW_L < QW < QW_U$
	$QW < QW_L$
	$QW > QW_U$
Queue Length (QL)	$QL_L < QL < QL_U$
	$QL < QL_L$
	$QL > QL_U$
Last Action	IncreaseServers
	DecreaseServers
	IncreaseCustomers
	DecreaseCustomers
Improvement	Yes
	No
IncreaseServers	True
	False
DecreaseServers	True
	False
IncreaseCustomers	True
	False
DecreaseCustomers	True
	False

One agent was used to inspect the model. The agent was asked to observe the model every twenty hours through four condition elements: the Customer Delay index, the Queue Length and Queue Wait index for the customers, and the Server Utilization Index. The observation interval was set at twenty hours to give the model time to stabilize and provide consistent values. The agent was also instructed to remember the last action taken upon the model and decide if the last action made any improvement to the model in terms of unit cost. These observations required an additional two condition elements for the agent. For this model, the cost of the shovel was set at $300/hr and the cost of the trucks at $150/hr to reflect a 2 to 1 cost ratio. There were four effect modeling elements for the model that simply acted if the 'IncreaseCustomers', 'DecreaseCustomers' 'IncreaseServers' or 'DecreaseServers' nodes met or exceeded their threshold belief of 0.8. Two of these effect modeling elements would not be required by the agent, as the server, or shovels, number was fixed at one. After 100 hours of simulation time and five observation periods, the agent had found that the preferred number of trucks for this operation was six. This is in agreement with the results obtained from the three previous studies (Carmicheal 1987; McCabe 1997; Van Tol 2000). The observations are provided in Table 2-3. The unit costs are provided for these observation intervals in Table 2-4. It should be noted that these results are provided for validation of this example. In actuality the results for the agent's decisions during runtime are not presented explicitly to the modeler.

2.6 Chapter Summary

This chapter outlined the process involved in developing the first agent for this research. The three criteria that had a significant impact on the maturity of the agent from a concept to prototype were the technical agent definition, the expectations of the agent for this research, and the simulation modeling environment in which the agent was to reside. The resulting agent structure made use of three distinct modeling elements: the parent agent modeling element, the condition modeling element, and the effect modeling element. The parent modeling element is the heart of the agent structure and provides the agent with all of its behaviors and abilities. The condition and effect modeling elements were designed to be the means by which the modeler would instruct the agent to define its agenda. The condition elements provide the modeler with the ability to instruct the agent as to what items of concern the agent was to observe. The effect elements were designed to allow the modeler to outline to the agent what possible actions the agent could carry out. The prototype agent described in this chapter was then demonstrated and validated in the evaluation of an earthmoving simulation model where the agent was employed to address resource allocation concerns.

Table 2-3. Agent Observations

Obs #	Sim Time	CD	SU	QW	QL	Last Action	Improvement	Action Taken
Initial Setting	0	Unobserved	Unobserved	Unobserved	Unobserved	Unobserved	Unobserved	None
1	20	$CD < CD_U$	$SU < SU_L$	$QW_L < QW < QW_U$	$QL_L < QL < QL_U$	IncreaseCustomers	No	None
2	40	$CD < CD_U$	$SU < SU_L$	$QW_L < QW < QW_U$	$QL_L < QL < QL_U$	IncreaseCustomers	No	Add
3	60	$CD < CD_U$	$SU < SU_L$	$QW_L < QW < QW_U$	$QL_L < QL < QL_U$	IncreaseCustomers	Yes	Add
4	80	$CD < CD_U$	$SU < SU_L$	$QW_L < QW < QW_U$	$QL_L < QL < QL_U$	IncreaseCustomers	No	None
5	100	$CD < CD_U$	$SU < SU_L$	$QW_L < QW < QW_U$	$QL_L < QL < QL_U$	IncreaseCustomers	No	None

Table 2-4. Model Properties at Observations

Obs #	Sim Time	Trucks In Model	Last Unit Cost	Current Unit Cost	Action Taken
Initial Setting	0	4	$ 0.00	$ 0.00	None
1	20	4	$ 0.00	$ 62.94	None
2	40	4	$ 62.94	$ 62.07	Add
3	60	5	$ 62.07	$ 55.41	Add
4	80	6	$ 55.41	$ 56.34	None
5	100	6	$ 56.34	$ 58.53	None

Chapter 3: Agent Definition and Applications

3.1 Introduction

The previous chapter introduced the agent structure developed for employment within a simulation modeling environment to address construction engineering and management issues. This chapter further explores the agent structure developed with an emphasis on two main areas. The first area is in regards to the agent's intelligence facilities. The developed agent structure is able to accommodate different forms of artificial intelligence for the agent. This ability will be demonstrated through the introduction of three additional agents, each with a different type of intelligence: neural network, algorithms for intelligence, and rule based intelligence faculty. The second area of focus further explores the function and characteristics of the agent in the simulation modeling environment by way of three additional example applications. The examples highlight the agent's ability to perceive and react to its environment, the agent's ability to make use of several forms of intelligence reasoning, and the agent's mobility that allows it to move from one environment to the next. The concepts presented in this chapter will then be expanded in the following chapter to develop the embedded agent simulation modeling framework.

This chapter is structured as follows. Section 3.2 will delve into the agent's intelligence faculties and how the intelligence chosen is influenced by the role the agent is to play within its environment Sections 3.3, 3.4, and 3.5 will present the neural network, algorithm, and rule-based agents respectively. Section 3.6 will provide three agent application examples adding further insight to the agent structure developed. Section 3.7 will conclude the chapter.

3.2 Intelligence and the Agent Structure

When approaching a problem with an artificial intelligence (AI) solution, the nature of the problem will often dictate the type of AI to use. It is for this reason the agent structure has been developed to allow the agent to possess different types of intelligence. Referring back to Figure 2-1, it can be seen that the agent's intelligence is a mechanism within the agent that processes observations to determine if any actions should be carried out. This structure does not mandate a specific type of reasoning, but indicates that some sort of process is required to allow the agent to realize the relevance of the information being received and how this relates to its agenda. This structure therefore allows for other forms of agent intelligence beyond that of belief networks. The following sections will describe three additional types of intelligence faculties that have been developed during the course of this research.

3.3 Neural Network Agent

The neural network agent is, as its name suggests, an agent with the ability to reason using feed-forward back-propagation neural networks. The agent was developed with the aid of NeuroWindows Neural Network Dynamic Link Library to handle the neural network calculations and a Microsoft Access Database file for data storage. Since the data were stored in a database file, a Visual Basic program was developed for manual data entry. Details regarding this program can be found in Appendix 2. In addition to the manual input of data, an observation assistant was developed with the ability to make observations and write information directly to the database file in an automated fashion.

3.3.1 Feed-Forward Back-Propagation Neural Networks

The field of neural networks is a very established, broad, and well-studied research area. The following is a brief introduction to one small area of this topic concerning feed-forward back-propagation neural networks. This introduction is intended to give insight into what is occurring within the constructs of the neural network agent presented in the following section. If further information is required regarding neural networks, reference can be made to Russell and Norvig (1994) or Ripley (1996).

In general, neural networks can be described as analytical pattern recognition tools that mimic the functions of neurons in the human nervous system. In the human nervous system billions of neurons work together in a large network to conduct electrical messages throughout the human brain and body. As shown in Figure 3-1, the human neuron is composed of a cell body with branched protrusions, called dendrites, and an axon. When a neuron functions in the human body an electrical impulse is received through the dendrites at the cell body and is transmitted through the axon. The neuron responds to the electrical impulse in either an excitatory or inhibitory manner by amplifying or diminishing the electrical impulse respectively across the synapse to the next neuron. The excitatory and inhibitory actions of the neurons determine how the electrical messages are conducted through the large network of nerves in the human body. Thus, if the brain were to initiate an electrical message instructing the left hand to open, the neurons in the body would produce excitatory responses to the nerves leading towards the left hand and inhibitory responses towards the other nerves. It is this process that artificial neural networks mimic.

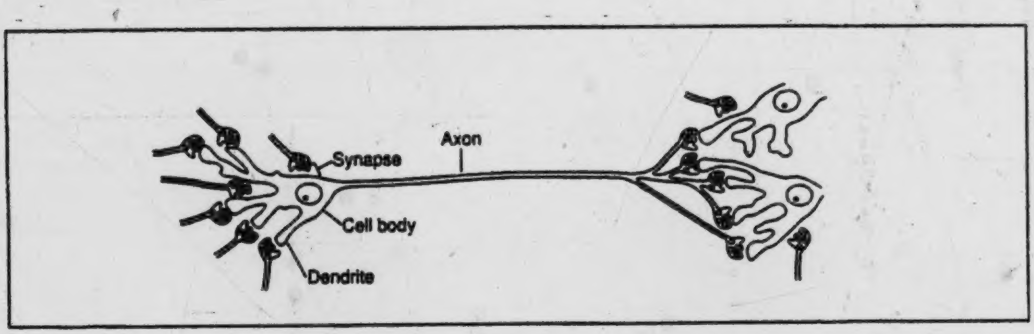


Figure. 3-1 Illustration of Neuron (AbouRizk 1993)

Artificial feed forward back propagation neural networks are constructed in a manner similar to that shown in Figure 3-2. Figure 3-2 shows a typical network constructed with four distinct components: the input layer, hidden layer(s), output layer, and bias node. The concept governing the neural network is that input variables are entered into the nodes found in the input layer and the nodes found in the hidden layer(s) mimic the biological process of receiving messages and producing excitatory and inhibitory responses to other nodes until results are obtained at the nodes found in the output layer. The single bias node placed in the network is used to shift the input values.

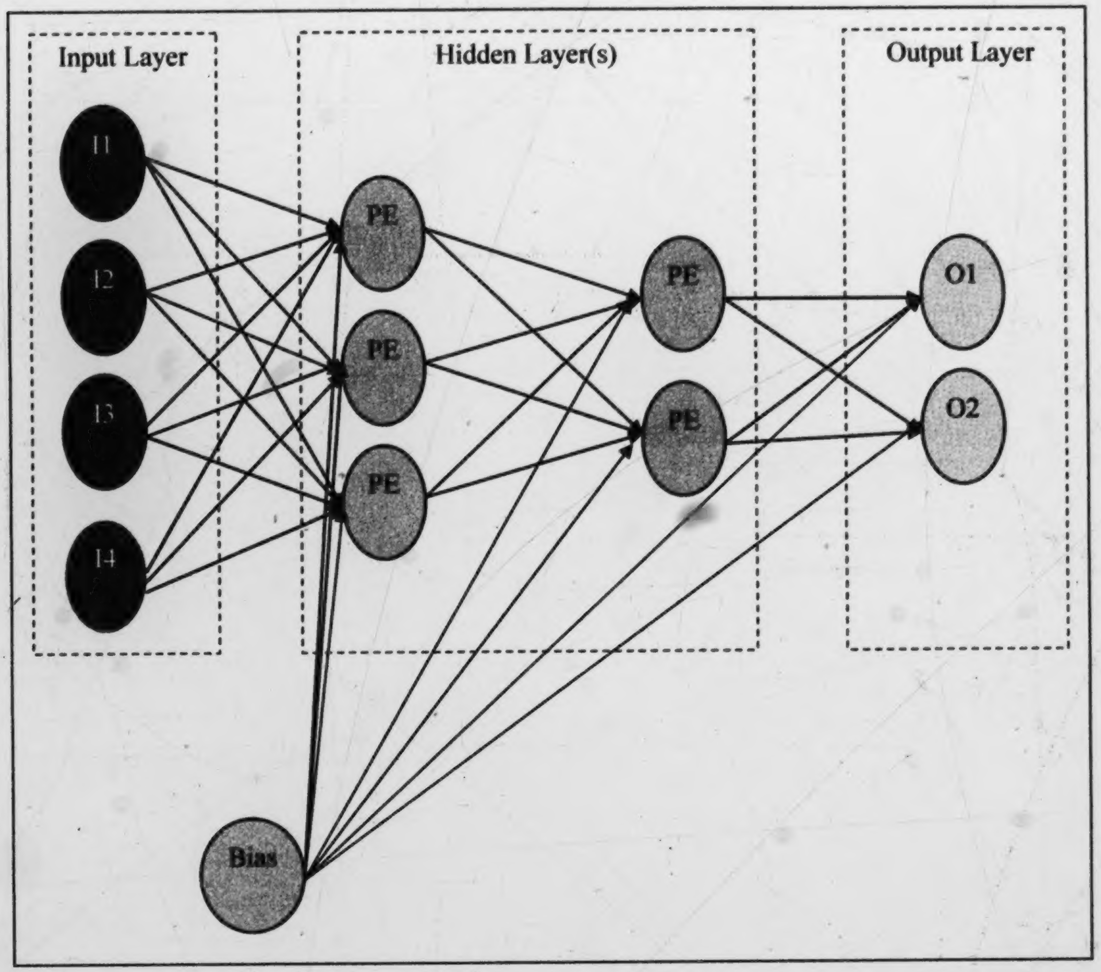

Input Layer
Hidden Layer(s)
Output Layer
I1
I2
I3
I4
PE
PE
PE
PE
PE
O1
O2
Bias

Figure. 3-2 Illustration of a Neural Network

Similar to the human body, these networks must be trained so that they function properly. This training is similar to the process involving a small child learning how to walk. The process is a developmental one as at first the child must acquire control over the movement of its legs, figure out the compliment of movements required for an awkward form of walking, and finally refining the coordination of the movements so that walking is achieved.

The child's command of walking develops as the network of neurons in the child's body begin develop and understand what electrical impulses should be excited or inhibited to allow walking to occur. This process is reflected in feed-forward back-propagation neural networks through two operations called training and recall.

Training is the learning process where the network figures out how it should function in regards to the excitatory and inhibitory responses for the nodes in the network. This process is done using test data that are able to compare the results of the network to actual output data. Learning requires a significant amount of historical data, which contains both the input and output variable values. The data are sectioned into training and test data. The split is often 80% of the data used for training data with the remaining 20% of the data used as test data. The training data are used to iteratively carry out the feed-forward back-propagation process where data are fed forward through the network, variances are calculated at the output nodes, and corrections to the network are back-propagated through the network from the output layer nodes. Once the network appears to be functioning to an acceptable error margin using the training data, the test data are introduced to the network to observe how the network reacts to new data. If the network is functioning satisfactorily, it is considered to be trained. If not, changes may be made to the network, in terms of the number of hidden layers and number of nodes within those layers, and the training is continued. Recall is the process where a trained network is used to provide confident output data based on input data entries with unknown outputs.

The nodes found in the network are the mathematical representation of biological neurons. These nodes are known as perceptrons or processing elements and are illustrated in Figure 3-3. The neuron is represented mathematically with weighted input arcs, the perceptron, and output arcs, which become the input arcs for successive perceptrons. Conceptually the artificial neuron functions by determining the total magnitude of its incoming "impulse" from all its input neurons and then sends out a fitting impulse to succeeding neurons in response. This response is determined through a transfer function. The magnitude of the incoming impulse is determined by a summation of all the incoming impulses for the perceptron, multiplied by the weight of the arc through which the impulse arrived. This summation is then applied to the transfer function, which calculates the responding impulse to send out. Transfer functions can vary, but a very simple transfer function would behave as follows: if the summation of impulses multiplied by their weights is greater than 0.3, then the outgoing impulse will be 1; otherwise, the outgoing impulse will be 0.

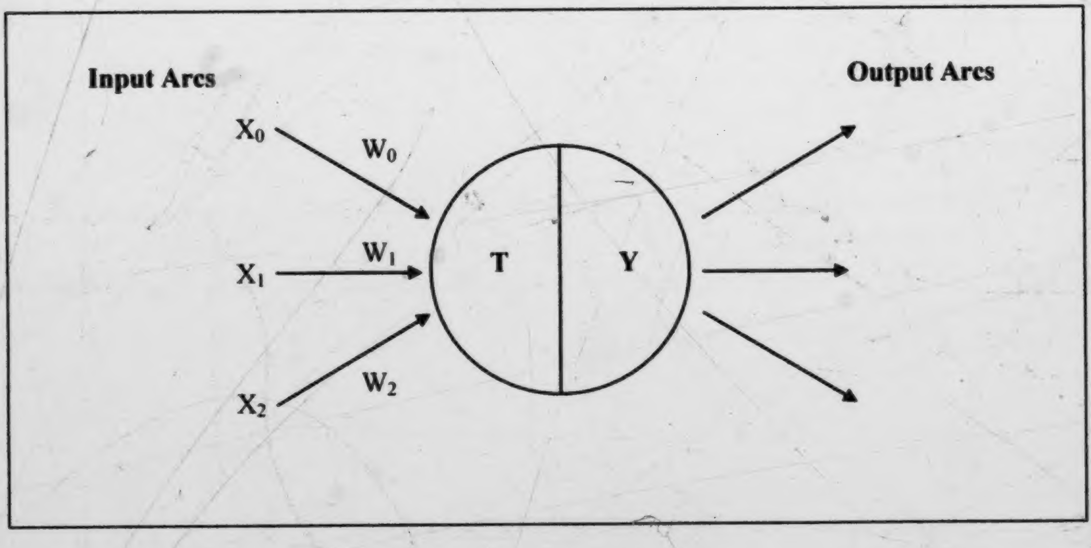


Figure. 3-3 Illustration of a Perceptron

The first step in training a feed forward back propagation neural network is to transform the data to [0,1] so that the data can be handled by the network. This can be achieved by applying equation 3.1 to the data. The next step is to randomly assign weighting to all of the arcs within the network. During network training, it is the weights of the arcs that will be adjusted during the error correction process. The next step is to process information through the network using equations 3.2 and 3.3. Equation 3.2 is used to calculate the total input value for the perceptron. Equation 3.3 is used to determine the output values for the perceptrons using the transfer function.

$$X_i = \frac{D - Min}{Max - Min} \qquad (3.1)$$

Where:

X_i = Transformed data value

D = Original data value

Min = Minimum value found in data

Max = Maximum value found in data

$$I_j = \sum_i W_{ij} X_i \qquad (3.2)$$

Where:

I_j = Summated input value for perceptron j

W_{ij} = Weight for arc from perceptron i to perceptron j

X_i = Output value from perceptron i

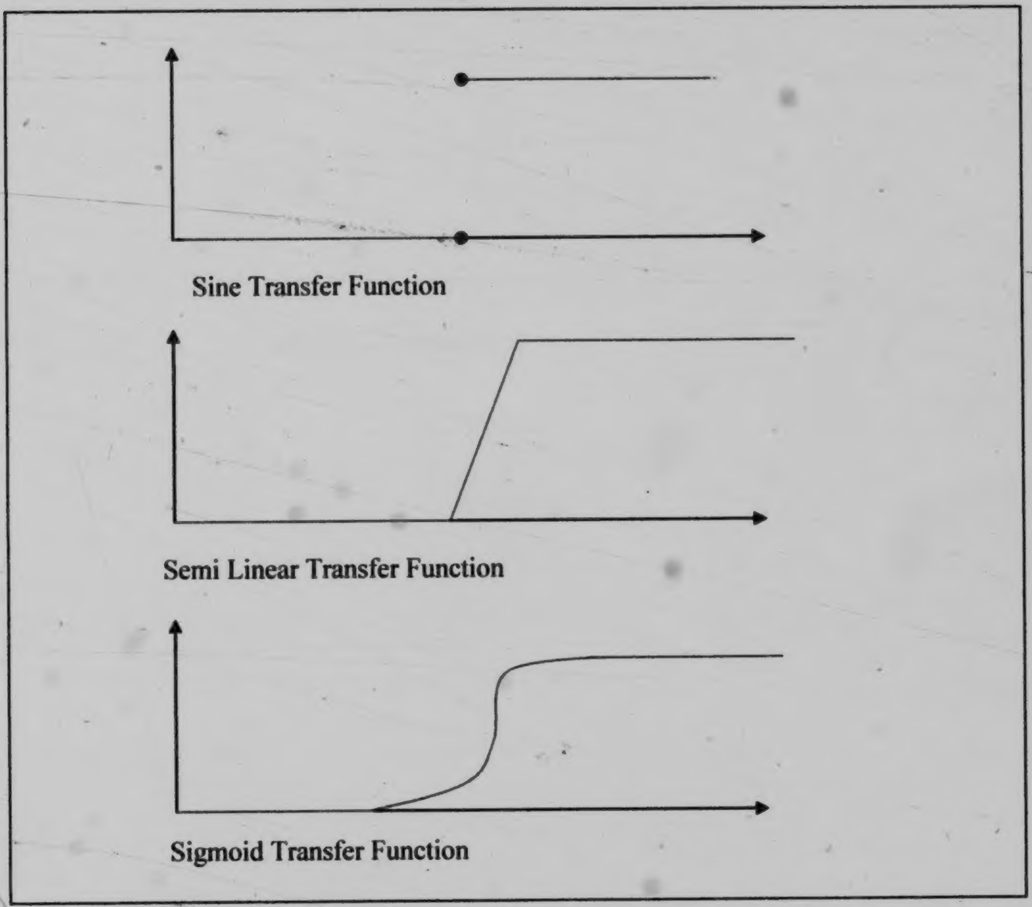
$$X_j = F(I_j) \tag{3.3}$$

Where:

X_j = Output for perceptron j

$F(I_j)$ = Transfer function

As mentioned previously there are several types of transfer functions. Figure 3-4 illustrates three examples of transfer functions. Transfer functions are mathematical functions that can be used to determine whether an inhibitory or excitatory response should be given by the perceptron based on the magnitude of the input value. In the case of the sine transfer function, the transfer function is discontinuous and provides an all or nothing response. The other two transfer functions provide varying degrees of response based on the magnitude of the input value.



Sine Transfer Function

Semi Linear Transfer Function

Sigmoid Transfer Function

Figure 3-4. Examples of Perceptron Transfer Functions

Once the information has been passed through the network, a global error calculation is made at the output layer perceptrons using equation 3.4. This equation determines an error value based on the variance between the expected and calculated output values. Once these values are calculated, an adjustment can be made to the arcs leading to the output perceptrons using equations 3.5 and 3.6. Once the weights of the arcs have been updated, calculations can be made to determine the local errors occurring at each of the preceding perceptrons using equation 3.7. This will allow for the updating of the remaining arcs in the network using equations 3.5 and 3.6 again. If desired, a momentum term can be added to equation 3.5 to improve the time required when training the network as shown in equation 3.8. The process of inputting information forward though the network and propagating the global and local errors back through the network is iteratively carried out until an acceptable average global error value is obtained. This process involves repeatedly applying the data vectors found within the training portion of the test data. Once the network has been found to be providing acceptable output values the remaining 20% of the test data are applied to the network to observe how well the network responds to new data and ensures that the network is not over-trained or generalized to the training test data.

$$\delta_j = (t_j - X_j)F'(I_j) \tag{3.4}$$

Where:

δ_j = Global error calculated at output perceptron j

t_j = True (or actual) output value expected at output perceptron j

X_j = Calculated output value expected at output perceptron j

$F'(I_j)$ = Derivative of transfer function at perceptron j

$$W_{ij} = W_{ij} + \Delta W_{ij} \tag{3.5}$$

Where:

W_{ij} = Weight for arc from perceptron i to perceptron j

ΔW_{ij} = Change of weight for arc from perceptron i to perceptron j

$$\Delta W_{ij} = \eta \delta_j X_i \qquad (3.6)$$

Where:

ΔW_{ij} = Change of weight for arc from perceptron i to perceptron j

η = Gain term or learning step size (usually 0.5)

δ_j = Calculated error at perceptron j

X_i = Calculated output value for perceptron i

$$\delta_i = F'(I_i) \sum_n \delta_{jn} W_{ijn} \qquad (3.7)$$

Where:

δ_i = Calculated error at perceptron i

$F'(I_i)$ = Derivative of transfer function at perceptron i

δ_{jn} = Calculated error at perceptron j,n

W_{ijn} = Weight for arc from perceptron i to perceptron j,n

n = Number of succeeding perceptrons j attached to perceptron i

$$W_{ij(n+1)} = W_{ij(n)} + \eta \delta_{jn} X_{jn} + \alpha (W_{ij(n)} - W_{ij(n-1)}) \qquad (3.8)$$

Where:

$W_{ij(n+1)}$ = Future weight for arc from perceptron i to perceptron j

$W_{ij(n)}$ = Current weight for arc from perceptron i to perceptron j

$W_{ij(n-1)}$ = Previous weight for arc from perceptron i to perceptron j

η = Gain term or learning step size (usually 0.5)

δ_{jn} = Current calculated error at perceptron j

Xj_n = Current calculated output value for perceptron j

α = Momentum term between 0 and 1 (usually 0.9)

3.3.2 Neural Network Agent

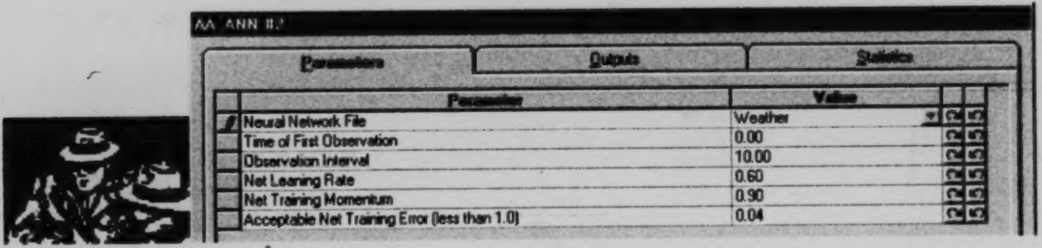
The neural network agent was developed following the agent structure and process presented in chapter 2. The only difference between the neural network agent and the belief network agent presented in the previous chapter is the intelligence faculty used by the agent. This difference gives cause to different applications for the two agents. The belief network agent can be applied to problems that are best suited to belief networks, such as problems that rely on expert opinion, deal with varying and missing data, and/or do not have a large amount of historical data from which solutions can be extrapolated. The neural network agent makes use of a neural network for its intelligence making it best suited to problems with structured input vectors and large amounts of historical data from which a network can be trained. While neither agent should be considered superior over the other based on their intelligence faculty, it should be emphasized that an agent's intelligence should reflect the nature of the problem being addressed.

As mentioned previously, the neural network agent's structure is the same as that presented in chapter 2, with an agent element and collection of condition and effect child elements.

Since a neural network is rigidly structured with regard to its inputs and outputs, when the modeler selects the actual neural network to be used by the agent, the agent will create and place its condition and effect child modeling elements for the modeler. The agent does this by creating one condition element for each input variable for the network and one effect element for each output variable from the network.

3.3.2.1 The Agent Element

The neural network parent agent modeling element has six modeling parameters, as shown in Figure 3-5. The first parameter, "Neural Network File", allows the modeler to select a neural network from a dropdown list of neural networks stored in a Microsoft Access database file. The "Time of First Observation" and "Observation Interval" parameters are exactly the same as for the neural network agent as for the belief network agent, and can be defined by linking the parameters or providing numerical values to define the agent's observation schedule. The remaining variables are used to instruct the agent how it should carry out its learning process when training a network. The "Net Learning Rate" parameter allows the user to define the learning rate found in equation 3.6. The "Net Training Momentum" refers to α, found in equation 3.8. The "Acceptable Net Training Error (less than 1.0)" parameter refers to the acceptable error level to which the network should be trained.

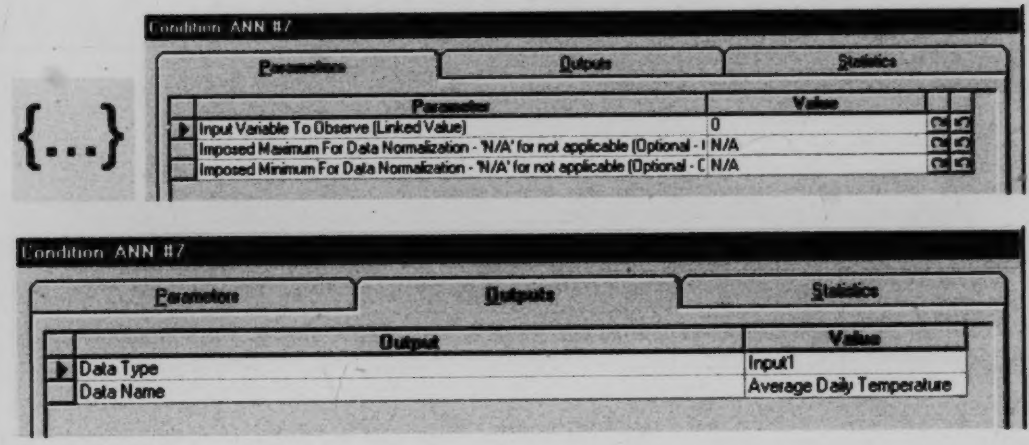


Parameter	Value
Neural Network File	Weather
Time of First Observation	0.00
Observation Interval	10.00
Net Leaning Rate	0.60
Net Training Momentum	0.90
Acceptable Net Training Error (less than 1.0)	0.04

Figure 3-5. Agent Element

3.3.2.2 The Condition Element

The condition elements are defined by the agent and modeler. The agent is responsible for placing the condition elements as its children and referencing these elements to input perceptrons in the network. The condition elements are accessible to the modeler through the "Outputs" tab of the element properties window as shown at the bottom of Figure 3-6. Using Figures 3-5 and 3-6 as examples, it can be seen that the condition element in Figure 3-6 references the "Average Daily Temperature" input perceptron within the "Weather" neural network. The three parameters for the condition element shown in Figure 3-6 are used to instruct the agent as to what information must be observed for the referenced perceptron. The "Input Variable to Observe (Linked Value)" is where the modeler instructs to the agent as to what is to be observed. The "Imposed Maximum For Data Normalization - 'N/A' for not applicable (Optional - Can be done based on training data)" and "Imposed Minimum For Data Normalization - 'N/A' for not applicable (Optional - Can be done based on training data)" parameters are provided to allow the modeler to explicitly define the maximum and minimum values to which the observation and training data referenced to this condition element should be normalized to. The modeler can define these maximum and minimum values for the agent or leave the agent to determine the maximum and minimum values from the data and normalize the[illegible] using equation 3.1.



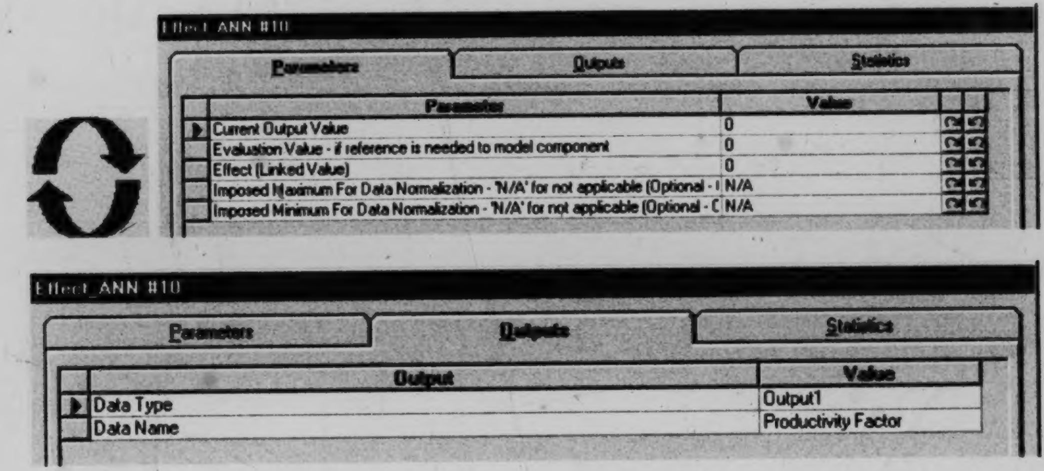
	Parameter	Value
▶	Input Variable To Observe (Linked Value)	0
	Imposed Maximum For Data Normalization - 'N/A' for not applicable (Optional - (	N/A
	Imposed Minimum For Data Normalization - 'N/A' for not applicable (Optional - (	N/A

	Output	Value
▶	Data Type	Input1
	Data Name	Average Daily Temperature

Figure 3-6. Condition Element

3.3.2.3 The Effect Element

The effect element is again created by the agent and identified to the modeler in the "Outputs" tab of the agent's properties. The "Current Output Value" parameter displays the current output value for the referenced perceptron from the neural network evaluation. The "Evaluation Value - if reference is needed to model component" is a parameter that allows the modeler to instruct the agent to compare this value to other information in order to determine its actions. "Effect (Linked Value)" is the parameter where the modeler instructs the agent as to what actions can be carried out. The "Imposed Maximum For Data Normalization - 'N/A' for not applicable (Optional - Can be done based on training data)" and "Imposed Minimum For Data Normalization - 'N/A' for not applicable (Optional - Can be done based on training data)" parameters serve the same purpose as their counterparts in the condition element by providing the modeler with the ability to explicitly define the maximum and minimum values for normalization.



Parameter	Value
Current Output Value	0
Evaluation Value - if reference is needed to model component	0
Effect (Linked Value)	0
Imposed Maximum For Data Normalization - 'N/A' for not applicable (Optional - I	N/A
Imposed Minimum For Data Normalization - 'N/A' for not applicable (Optional - C	N/A

Output	Value
Data Type	Output1
Data Name	Productivity Factor

Figure 3-7. Effect Element

3.3.2.4 Observation Process

The observation process carried out by the neural network agent is very similar to that of the belief network agent, with the agent looking to its condition elements for observation information, evaluating the information through its intelligence faculties, and then looking to its effect elements for actions that should be carried out. The main difference between the two agents is that the neural network agent must construct and train a neural network before evaluating the observation information.

The first step the agent carries out before evaluating information is to construct and train the neural network. The agent knows the number of input and output perceptrons to place in the input and output layers. The agent also knows that a single bias perceptron has to be placed within the network. The only unknown for the agent is the construction of the hidden layer. To simplify this process the agent has been designed to use only one hidden layer in the network. The number of nodes in that layer is determined using equation 3.9. Once the network is constructed, the agent carries out the learning process for the network until the average global error in the network is within the tolerance range specified by the modeler in the parent agent element parameters. The data used for the learning process are provided to the agent via a Microsoft Access database file. Once the learning process is done, the agent carries out the recall process using the observation data identified in the agent's condition elements. The results of the recall process are then applied to the effect elements so that any valid actions can be carried out.

$$Hidden = \frac{1}{2}(Input + Output) + \sqrt{Data\mathrm{Re}cords} \qquad (3.9)$$

Where:

Hidden = Number of hidden layer perceptrons

Input = Number of input layer perceptrons

Output = Number of output layer perceptrons

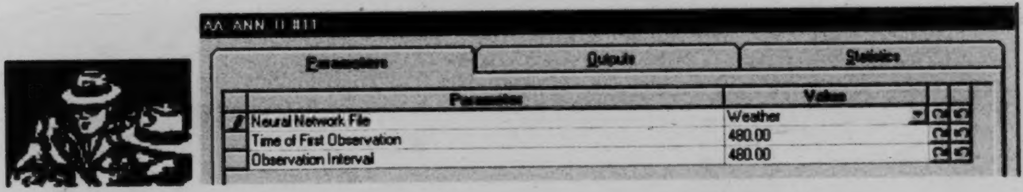
DataRecords = Number of training vectors in database

3.3.2.5 The Observation Assistant Element

As mentioned in the beginning of the description of the neural network agent, the agent has an observation assistant that is able to make observations within a simulation environment and record information directly to the database file that the agent uses for constructing and training its neural networks. The observation element consists of only two model elements: a parent observation assistant modeling element containing a collection of child observation elements in a fashion similar to the agent structure. The parent observation assistant modeling element is presented in Figure 3-8. This element has only three parameters. The "Neural Network File" parameter is a drop down list that indicates to the assistant what neural network within the database file the information is to be written. The selection of the neural network file will also provide the assistant with the information required to create the required child observation elements. The "Time of First Observation" and "Observation Interval" parameters define as observation schedule for the assistant in a manner similar to the agent's observation schedule.

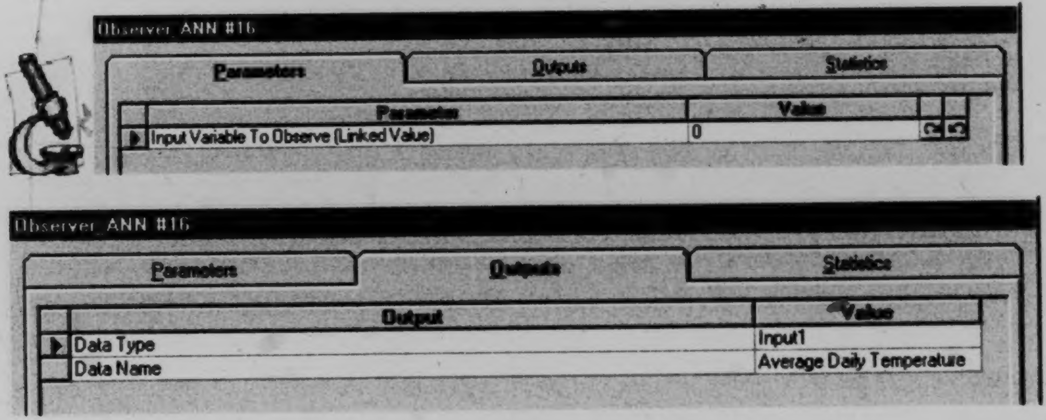
The observation element is created by the parent observation assistant modeling element and identified to the modeler in the "Outputs" tab of the element properties, as shown in Figure 3-9. The observation element only has one parameter, which identifies what information is to be captured and written to the database.

The recording process for the assistant is straightforward. The process is initiated with the identification of an observation interval. The assistant looks to its child elements to determine what information must be written to the database and then writes the information to the database. The benefit the assistant provides for the neural network agent is the ability to provide data to the database file from which the agent constructs and trains the neural networks. This allows the agent to make use of simulated historical data collected from the simulation environment in which the agent has been placed.


AA ANN II #11
Parameters
Outputs
Statistics

	Parameter	Value
	Neural Network File	Weather
	Time of First Observation	480.00
	Observation Interval	480.00

Figure 3-8. Observation Assistant Element


Observer ANN #16
Parameters
Outputs
Statistics
Observer ANN #16
Parameters
Outputs
Statistics

	Parameter	Value
	Input Variable To Observe (Linked Value)	0

	Output	Value
	Data Type	Input1
	Data Name	Average Daily Temperature

Figure 3-9. Observation Element

3.4 Algorithm Agent

The algorithm agent demonstrates the agent structure's ability to reason based on algorithms. The algorithm used to demonstrate this ability is a soil prediction algorithm developed by Ruwanpura (2001). In general, the soil prediction algorithm uses borehole log data to determine soil conditions at target locations. In terms of agent development, the algorithm provides the agent with the ability to predict soil conditions within a simulation modeling environment and, in turn, affect the simulation model based on these predicted soil conditions.

3.4.1 Soil Prediction Algorithm

The soil prediction algorithm developed by Ruwanpura (2001) predicts the presence of soil families at target locations based on borehole information for the area surrounding these target locations. A soil family can be described as a grouping of soil types at a target location described in the specific order in which they are found at the location. For example, a surfacial soil such as Saskatchewan gravel sand could overlay a glacial till formation, which overlays a bedrock formation. In this instance the soil family would be [Saskatchewan gravel sand, glacial till, bedrock]. In another location where the Saskatchewan gravel sand is absent, the family would be [glacial till, bedrock]. In a location where the glacial till overlays the Saskatchewan gravel sand, the family would be [glacial till, Saskatchewan gravel sand, bedrock]. The algorithm predicts a family for a target location using two equations to calculate the probability of the existence for a variety of families at the target location. The resulting soil family with the highest probability of being present at the target location is determined to be the soil family present.

The first step to determine the soil family present at the target location is the determination of the probability of co-existence (PCE) of particular soil families with other soil families in the area of study using equation 3.10. Basically, equation 3.10 calculates the PCE values by tallying the occurrences of the base family (soil family of concern) and the occurrences of the borehole family (soil family being compared against) in the area of study and applying those numbers to equation 3.10. For example, an area of study may have a borehole database containing 200 borehole logs. From the 200 borehole logs in the database, 10 of those logs have glacial till overlying bedrock and 50 of those logs have Saskatchewan gravel sand overlaying bedrock. The PCE value addressing the co-existence of glacial till overlaying bedrock given the presence of Saskatchewan gravel sand overlaying bedrock is calculated by dividing the 10 borehole logs by the sum of the 10 borehole logs and 50 borehole logs. The resulting PCE value is thus 10 divided by 60, or 0.167. One note in regard to equation 3.10, if the base family is being compared to itself, where the base soil would be the same as the borehole soil, the PCE value would be 1.0, not 0.5.

$$PCE_{Base,BHi} = \frac{F(Base)_{SoilFamily}}{(F(Base)_{SoilFamily} + F(BHi)_{SoilFamily})} \qquad (3.10)$$

Where:

$PCE_{Base,BHi}$ = The probability of the coexistence of the soil family found in borehole i in a given area

$F(Base)_{SoilFamily}$ = The number of occurrences of the base soil family in the database within a given distance F

$F(BHi)_{SoilFamily}$ = The number of occurrences of the soil family in borehole i existing in the database within a given distance F

The PCE values are calculated for all possible combinations of soil families being dependant on other soil families found in the area of study. Once this task is completed, the weight factor for each particular soil family being present at the target location is calculated using equation 3.11. Equation 3.11 determines the weight factor for each soil family by applying the individual PCE values to the distance between the boreholes used to calculate the respective PCE values and the target location. The concept of equation 3.11 is to provide a higher weight factor to those boreholes closest to the target location. The resulting family with the largest weight factor is deemed to be the predicted soil family present at the target location.

$$WF_{T,SoilFamily\ A} = \frac{\sum_{BH=1}^{n} \frac{1}{d_{BH}} PCE_{BH,SoilFamily\ A}}{\sum_{BH=1}^{n} \frac{1}{d_{BH}}} \tag{3.11}$$

Where:

$WF_{T,SoilFamilyA}$ = Weight factor for a particular soil family at a target point

BH = Borehole record number in database for the area of consideration

n = Number of boreholes in database for the area of consideration

d_{BH} = Distance from borehole to target location

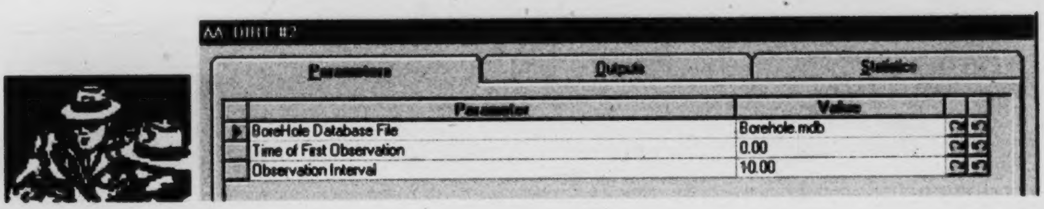
$PCE_{BH,SoilFamilyA}$ = Probability of the coexistence of soil family A in the area of consideration

3.4.2 Soil Prediction Agent

The soil prediction agent follows the agent structure outlined in chapter two and, in a similar fashion to the neural network agent, looks for information from a Microsoft Access database file external to the simulation modeling environment. The borehole database to which the agent refers is described in Appendix 1. The following section will describe the soil prediction agent.

3.4.2.1 The Agent Element

The agent parent element shown in Figure 3-10 has three parameters that the modeler must complete to define the agent. The "BoreHole Database File" parameter is where the modeler instructs the agent as to what database file it should use. The "Time of First Observation" and "Observation Interval" parameters are used to define the observation schedule for the agent and are the same as their counterparts in the agents described previously.

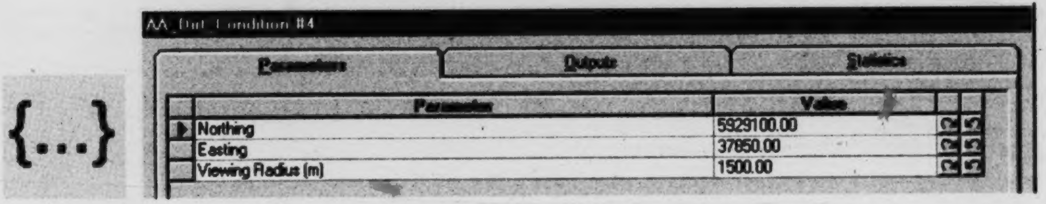


Parameter	Value
BoreHole Database File	Borehole.mdb
Time of First Observation	0.00
Observation Interval	10.00

Figure 3-10. Agent Element

3.4.2.2 The Condition Element

The condition element has three element parameters and is shown in Figure 3-11. The "Northing" and "Easting" parameters are used to define the target location of where the soil family is to be predicted. These parameters are to be defined with northing and easting values relative to the northing and easting references for the boreholes found in the database file. These parameters can also be linked to allow for changing target locations. Using a tunnel construction example, the northing and easting values could be linked to equations that determine the northing and easting of the tunnel face based on an equation of the designed tunnel alignment and the distance excavated by the tunnel boring machine. The "Viewing Radius (m)" parameter defines the area of study surrounding the target location that the agent should consider when using the soil prediction algorithm.



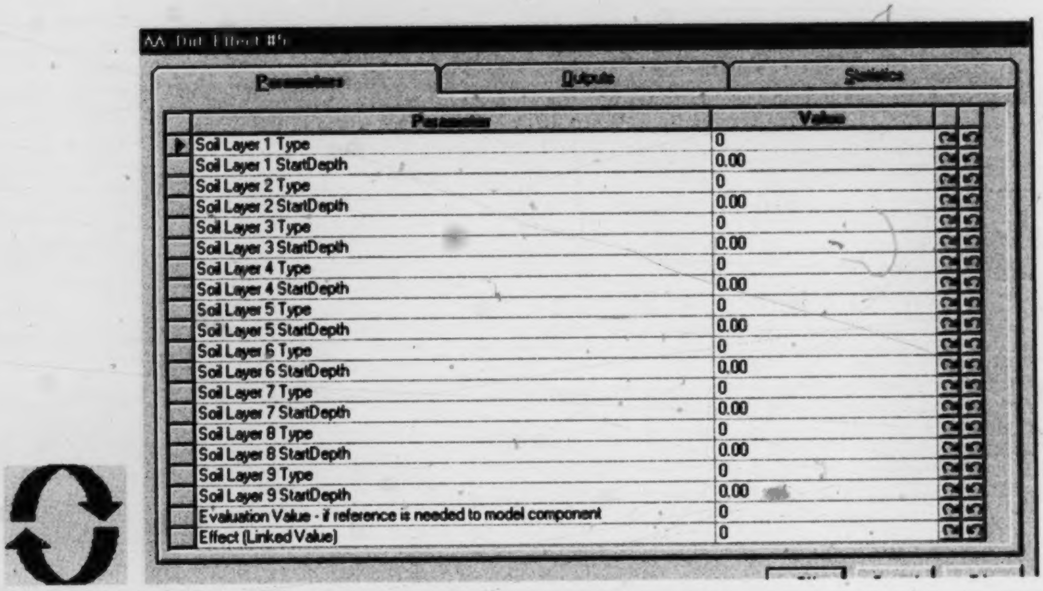
Parameter	Value
Northing	5929100.00
Easting	37850.00
Viewing Radius (m)	1500.00

Figure 3-11. Condition Element

3.4.2.3 The Effect Element

The effect element has several modeling element parameters, as shown in Figure 3-12. The bulk of these parameters are used to list the soil present in the artificial borehole predicted and the depths at which these soil layers start. These parameters are not of concern to the modeler. They are used to display the results of the soil prediction process. This leaves

"Evaluation Value - if reference is needed to model component" and "Effect (Linked Value)" parameters requiring attention by the modeler. The "Evaluation Value - if reference is needed to model component" parameter provides a location for the modeler to instruct the agent to carry out further observations and evaluations if required. The "Effect (Linked Value)" parameter is where the modeler instructs the agent as to what actions can be carried out and when these actions should be carried out.



Parameter	Value
Soil Layer 1 Type	0
Soil Layer 1 StartDepth	0.00
Soil Layer 2 Type	0
Soil Layer 2 StartDepth	0.00
Soil Layer 3 Type	0
Soil Layer 3 StartDepth	0.00
Soil Layer 4 Type	0
Soil Layer 4 StartDepth	0.00
Soil Layer 5 Type	0
Soil Layer 5 StartDepth	0.00
Soil Layer 6 Type	0
Soil Layer 6 StartDepth	0.00
Soil Layer 7 Type	0
Soil Layer 7 StartDepth	0.00
Soil Layer 8 Type	0
Soil Layer 8 StartDepth	0.00
Soil Layer 9 Type	0
Soil Layer 9 StartDepth	0.00
Evaluation Value - if reference is needed to model component	0
Effect (Linked Value)	0

Figure 3-12. Effect Element

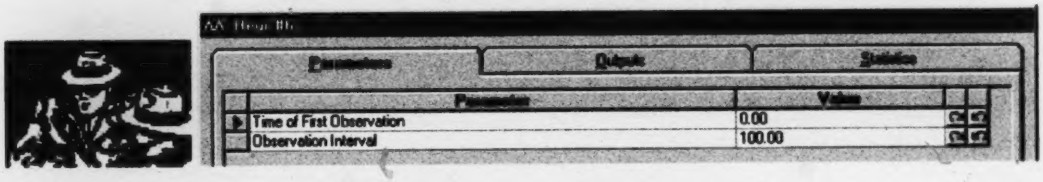
3.4.2.4 Observation Process

The observation process for the agent is generally the same process as is used for the two previous agents. The soil prediction agent identifies an observation interval, it looks to its condition elements to see what observations must be made, processes the information from the observations, and lastly looks to its effect elements for any valid actions.

The detailed observation process is initiated with the soil prediction agent looking to its condition element to determine where the target location is and to define the area of study surrounding the target location. Once this is done the agent looks in the borehole database in preparation for the soil prediction algorithm. The agent prepares itself by finding the boreholes in the area of study and extracting their information from the database file. The agent then begins applying the extracted data to the prediction algorithms. The agent then determines the predicted soil family for the target location. One item that the prediction algorithms do not address is the upper and lower elevations for the soil layers in the soil family. This problem was addressed by using simple geometry. The agent identifies each soil type in the predicted family and then finds the three closest boreholes with that soil type such that a triangular plane can be constructed and elevations determined. It should be noted that this method of determining elevation was done solely to reduce development time. The method of using simple geometry demonstrates the concepts presented, but can be replaced with more sophisticated methods such as digital terrain modeling. Once the agent has determined the elevations for the soil types, the agent has constructed an artificial borehole for the target location and the agent can then look to its effect elements for instruction on actions to be taken.

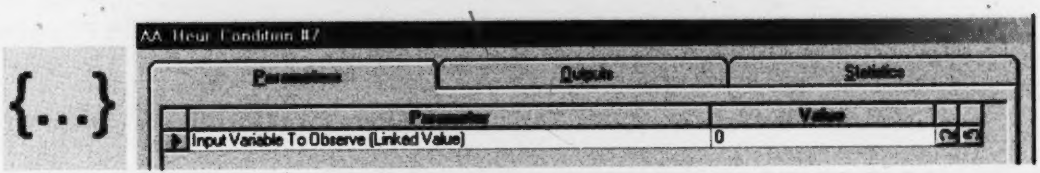
3.5 Rule-Based Agent

The rule-based agent consists of the basic agent structure with an if/then operation for the agent. The agent has no artificial intelligence reasoning faculties, but has the ability to make observations of its environment and take action based on these observations. The parent element is presented in Figure 3-13. The agent's observation schedule is defined using two parameters in the parent element. The condition element is presented in Figure 3-14. It has only one parameter where the modeler instructs the agent as to what observations are to be made. The effect element is presented in Figure 3-15. It has two parameters: the first parameter provides a means for the agent to make further investigations into its environment, and the second parameter defines the agent's actions. The agent observation process is very simple, as the agent looks to its condition elements for observations and then applies these observations to its effect elements to determine what actions should be taken.


AA Heur #6
Parameters
Outputs
Statistics

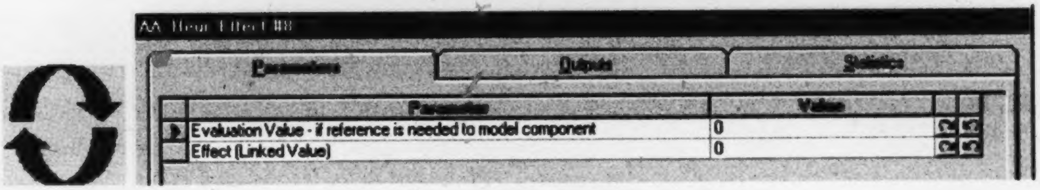
Parameter	Value
Time of First Observation	0.00
Observation Interval	100.00

Figure 3-13. Agent Element


AA Heur Condition #7
Parameters
Outputs
Statistics

Parameter	Value
Input Variable To Observe (Linked Value)	0

Figure 3-14. Condition Element


AA Heur Effect #8
Parameters
Outputs
Statistics

Parameter	Value
Evaluation Value - if reference is needed to model component	0
Effect (Linked Value)	0

Figure 3-15. Effect Element

3.6 Agent Examples

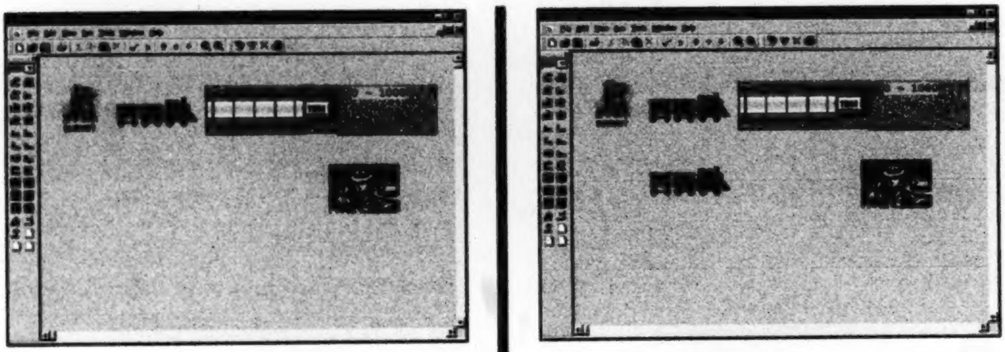
The following examples are provided to illustrate two points. First, the examples further demonstrate the agents developed during this research. Second, the examples demonstrate the applicability of the agents to their environments, highlighting the agents' mobility and transposability. The first example is presented in detail in Appendix 4 and uses a belief network agent to address tunnel construction concerns. The second example, which is presented in detail in Appendix 6, presents a neural network agent employed to address weather concerns for the construction of a bridge abutment. The last example uses the same agent from example 2 to address weather impact on an earthmoving operation. It is presented in detail in Appendix 7.

3.6.1 Example 1: Tunnel Construction

One of the concerns in tunnel construction is the balancing of the muck cars, used to haul material away from the tunnel excavation face, as the tunnel boring machine (TBM) progresses. As a project manager one of the key concerns during tunnel construction is to ensure that an adequate soil removal system is in place for the TBM so that it remains productive. The TBM is regarded as the primary production unit for the operation and, generally, if the TBM is not productive in its excavation then neither is the tunnel construction project. The facilitation of soil removal is done through the use of trains or muck cars. The trains simply haul soil away from the TBM to the removal shaft and back haul materials to the excavation face that are required for tunnel lining and rail line construction.

The balancing of the muck cars for the TBM is of greatest concern at the early stages of tunnel construction. When the excavated tunnel chainage is a short distance, so are the train cycle times. The short cycle times make it easy for a single train to balance off the TBM. Also aiding the single train in maintaining balance with the TBM at short distances is the requirement for support functions, such as lining installation, to be carried out at 1-metre advancements of the TBM. These support activities are carried out as the trains haul material and therefore, at shorter travel distances, consume the idle time experienced by the TBM when the trains are away, making it practical to only employ one train. The use of one train at the beginning of construction is a cost effective practice as it would not be considered cost-effective to employ two trains at a stage in construction when only one train is required. The question that remains, however, is when to employ the additional train? This question can be answered through the use of an agent observing a model of the construction process in a simulation environment.

To further improve tunneling efforts within the City of Edmonton, a tunneling template was developed that allowed the project personnel to explore the tunneling project artificially in a simulation environment (Mohammed 2002). The simulation template developed is comprehensive and accounts for many variables, such as soil types, labor impacts, shift schedules, and equipment efficiency. The first step in solving this problem is to create a model of the tunneling project using this template (Figure 3-16).



Prior to Simulation

Post Simulation

Figure 3-16. Tunnel Construction Simulation Screen Captures

Once the tunnel construction model has been developed, the next step is to employ an agent within the model to determine when the second train should be added. When the modeler creates the agent, the modeler should first determine and develop the intelligence required for the agent to carry out this problem. In this case, the project superintendent's opinion provides the information required to determine when the second train should be added. The superintendent implements a second train if the average waiting time for the trains at the TBM is greater than two minutes and if the waiting times are progressively getting longer. The superintendent also accounts for start up inefficiencies and will not add a second train to the project if the start up operations are not well coordinated due to the learning curve. This type of information leads the researcher towards the use of belief networks to capture the intelligence for this problem.

The belief network developed for the agent's use in solving this problem is found in Figure 3-17. Four nodes were used to construct this network and each node had a state of "yes" or "no". The node labeled "CurrentObsGTLast" was used to represent the superintendent's question as to whether the current waiting time was greater than the last waiting time. The superintendent's question of whether the current waiting time was greater than two minutes was represented using the node labeled "CurrentObsGT2." The node labeled "CountGT1" was used to represent the superintendent's question: "Are we past the start up learning effect?" Lastly, the "AddTrain" node was used to attain belief values of either "yes" or "no" in response to the question of whether another train needed to be added to the operation.

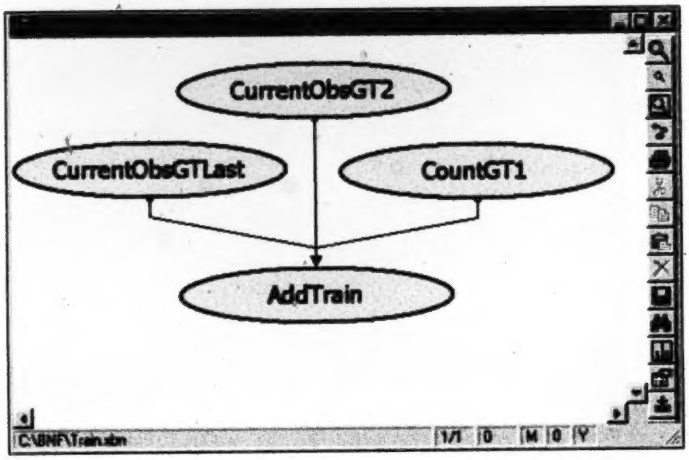


Figure 3-17. Train Belief Network

Once the intelligence for the agent has been determined, the next step involves instructing the agent as to what it should do though its agenda. Three condition elements and one effect element were required to define this agent's agenda. The three condition elements explained the agent's role, which was to observe the average waiting time at the TBM, the difference between the current waiting time and the last observed waiting time, and the number of cycles for the train(s). The information was used as input information for the belief network. The effect element instructed the agent to add a train to the simulation model if the resulting belief value for the state "yes" for the node "AddTrain" was above 0.75. The final step in creating this agent was to instruct the agent on how often it was to make its observations. In this case, the agent was asked to make observations every 240 minutes.

The resulting influence of the agent on the simulation model can be seen in Figure 3-16. The information regarding the agent's decision-making process is found in Table 3-1. Figure 3-16 shows two screen-captures. The first capture depicts the simulation model as it would be from simulation time 0 to 1679 minutes. The second capture is of the model from simulation time 1680 forward. Notice the two train modeling elements in the post-simulation screen capture. From these two screen captures it can be seen that the agent was able to change the topology of the model by adding another train modeling element in order to achieve the goal of maintaining a balance between the trains and the TBM excavation. Table 3-1 further reinforces the point that the agent was able to influence its environment through the management of waiting times at the TBM. Table 3-1 shows a progressively longer average waiting time at the TBM with an average waiting time of greater than two minutes at simulation time 1680 minutes. At 1680 minutes the second train was added and the average waiting times were reduced in future observations to tolerable levels. The resulting impact of this agent identified to the modeler that the second train should be added to the project at chainage 311 in order to maintain a balance between the trains and the TBM.

Table 3-1. Results for Agents Actions in Tunnel Construction Simulation Model

Observation	Sim Time (min)	Chainage (m)	Waiting Time (min)	Last Waiting Time (min)	Waiting Time Greater Than 2 Min.	Waiting Time Greater Than Last	Number of Obs. Greater Than 1	Action
5	960	299	1.824	1.732	No	Yes	Yes	None
6	1200	305	1.908	1.824	No	Yes	Yes	None
7	1440	311	1.989	1.908	No	Yes	Yes	None
8	1680	318	2.067	1.989	Yes	Yes	Yes	Add Train
9	1920	324	1.814	2.067	No	No	Yes	None
10	2160	330	1.587	1.814	No	No	Yes	None

3.6.2 Example 2: Abutment Construction

In the previous example, a belief network was chosen for the agent's intelligence based on expert opinion from a construction superintendent. In this example, a problem is presented that requires the use of an agent able to reason through neural networks. The problem looks at using an agent to apply the impacts of weather on a modeled bridge construction project. Wales (1994) developed a method to quantify the effects of weather on construction by way of a neural network that calculated a productivity factor applicable to a baseline productivity value to correct for weather conditions. The neural network Wales (1994) developed required the daily average temperature, the daily precipitation, and a seven day cumulative precipitation for an input vector and returned a productivity factor as output. As an example, if the daily average temperature was 14.7 degrees Celsius, the daily precipitation was 0.0

mm, and the seven day cumulative precipitation was 13.8 mm the network would produce a productivity factor of 1.18. The value of 1.18 would then be applied to the baseline productivity for that day to portray how actual construction progressed with respect to the weather conditions.

In regards to the problem at hand, the neural network agent was provided with a collection of historic data records containing the daily average temperature, the daily precipitation, the seven day cumulative precipitation, and the resulting productivity factor used to create the network developed by Wales (1994). The agent used this data to create a neural network, train that network, and use the trained network to make inferences regarding the simulation model being observed.

Figure 3-18 is a graphical representation of the validation problem used to demonstrate the network developed by Wales (1994). Wales had investigated the bridge abutment excavation activity within an activity-on-arrow diagram. The results obtained from Wales (1994) are presented in Table 3-2. From Table 3-2 it can be seen that the excavation activity was broken down into its individual shifts and the weather conditions were provided for each shift. The production factor was then determined and applied to each shift. The end result found that the 40 hour activity required 48.3 hours to complete when weather conditions were considered.

The reproduction of Wales's example will be used to further validate the agent. The weather conditions listed in Table 3-2 will be the same conditions used to test the agent's ability to adjust production in the simulation model. Thus, the agent will be presented with five days of different weather conditions. The first, second, and fourth days will consist of fair weather conditions and the productivity factor can be expected to be around 1.0. The third and fifth days will have poor conditions and the productivity factor can be expected to be less than 1.0.

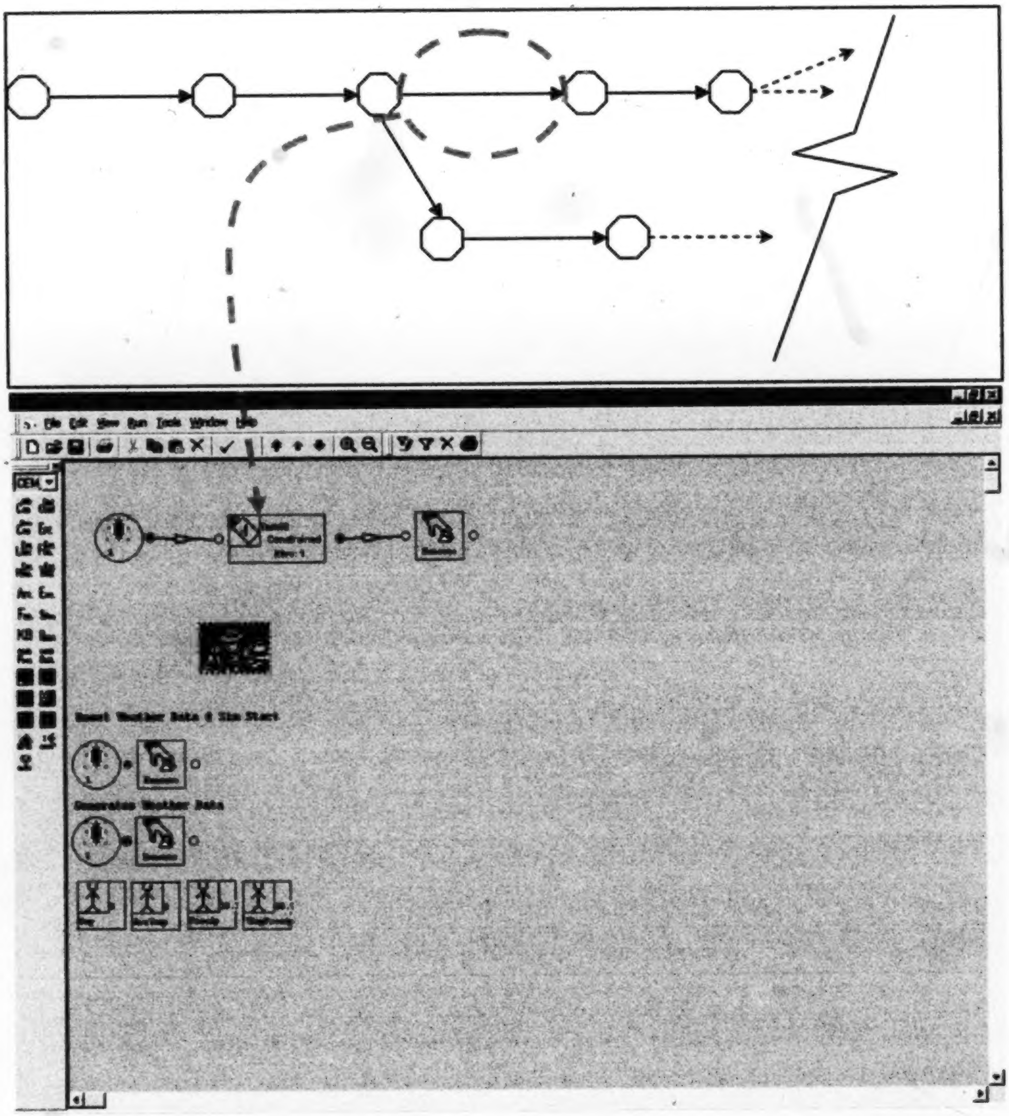

File Edit View Run Tools Window Help
Reset Weather Data & Sim Start
Generate Weather Data

Figure 3-18. Bridge Construction CPM Model and Simulation Abstraction

Table 3-2. Weather Data of Abutment Construction with Weather Effects (Wales 1994)

Day	Average Temperature (°C)	Precipitation (mm)	7-Day Cumulative Precipitation (mm)	Production Factor	Shift Length (Hours)	Total Work Hours
1	14.7	0.0	13.8	1.18	10	11.8
2	16.4	0.0	14.6	1.18	10	23.6
3	13.9	12.0	16.5	0.37	10	27.3
4	12.3	0.7	26.0	1.07	10	38
5	9.0	10.3	36.7	0.24	10	40.4

The model developed in the simulation modeling environment for this problem is presented in Figure 3-18. The single activity in the activity-on-arrow diagram is abstracted into a simulation model through a single task modeling element from the Common Modeling Template in *Simphony* (Hajjar and AbouRizk 2002). One slight modification was made to this element to enable segmentation of the activity into shifts. This element was originally constructed to schedule a single event in the simulation calendar such that the simulation modeling entity was delayed for the duration of the entire task. The task element was therefore changed so that it scheduled several shift-length delays for an entity until the activity was completed. This was done to give the agent the ability to change the productivity of an activity on a daily basis. The elements located below the agent in Figure 3-18 are used to represent the data source for the weather information. These elements contain the information for the five days of weather used in Wales' (1994) validation (Table 3-2). It should be noted that although the weather information was presented to the agent in this manner, the agent could also be asked to reference an actual weather database outside of the modeling environment.

The agent was embedded into the simulation environment with the parent agent given reference to the historical weather data records from which to construct a neural network, and instructed to observe the model at the beginning of every shift. The three condition elements for the agent contained the instructions to observe the daily average temperature, the daily precipitation, and the seven day cumulative precipitation. The single effect element instructed the agent to apply the productivity factor, determined by the neural network, to the upcoming shift's productivity.

Table 3-3 presents the results of the agent's influence on the model. One can see that this agent acted similarly to Wales' prediction (1994). The agent determined favorable productivity on the first, second, and fourth shifts and poor productivity on the third and fifth days. The agent's influence resulted in the planned 40 hour task requiring 48.9 hours to actually complete. This is 0.6 hours longer than Wales' prediction, which equates to a 1.17 % margin of error between the two results.

Table 3-3. Reproduced Results for Abutment Construction with Weather Effects

Day	Average Temperature (^oC)	Precipitation (mm)	7-Day Cumulative Precipitation (mm)	Production Factor	Shift Length (Hours)	Total Work Hours
1	14.7	0.0	13.8	1.1252	10.0000	11.2516
2	16.4	0.0	14.6	1.1359	10.0000	22.6101
3	13.9	12.0	16.5	0.4386	10.0000	26.9959
4	12.3	0.7	26.0	1.0118	10.0000	37.1140
5	9.0	10.3	36.7	0.3247	8.8888	40.0000

3.6.3 Example 3: Earthmoving Simulation

This example is intended to demonstrate the agent's transposability between simulation modeling environments. It also demonstrates the agent structure's mobility, which allows the agent to move freely within its modeling environment from the observation of one model to the next. To demonstrate this concept, the same agent used in example 2 was used to apply the same influence of weather conditions to a different model. In the current example, the agent from example 2 was asked to apply the influences of weather to an earthmoving model developed using the Special Purpose Simulation (SPS) Earthmoving Simulation template (Hajjar 1999).

The earthmoving model is presented in Figure 3-19. In this screen capture, it is visible that the same agent, along with its artificial weather database, has been embedded in the modeling environment with the earthmoving model. Through the child window of the earthmoving simulation, it is visible that the earthmoving model consists of a source location connected to a placement location and a dump location (for waste material) through a series of roads. The model is relatively easy to comprehend, as the model closely resembles an actual earthmoving operation. In the child window for the source location, one can see that there is a shovel modeling element, a material pile (soil) element, and additional elements required for statistics collection. The focus for the agent is the shovel modeling element. Similar to the previous example, where the agent was adjusting the productivity of the abutment excavation based on the weather, the agent will adjust the rate at which the shovel is able to load the trucks based on weather conditions.

Essentially, the agent in the previous example was copied from its previous environment and pasted into the new example's environment. The agent's condition elements remain the same as a result of the agent referencing the same weather data. The weather conditions in the previous example were used for this example in order to further demonstrate the agent's mobility and its ability to exert a similar influence over two different types of models. The only significant change was applied to the instructions found in the agent's effect element. The previous instructions asked the agent to apply the productivity factor to the Task element. In this example, however, the instruction was changed to have the agent apply the productivity factor to the shovel productivity rate. The observation schedule was also changed in order to enable the data collection of several truck cycles to demonstrate the agent's influence. Table 3-4 presents the loading times for the trucks over the duration of the agent's influence upon the model. It should be noted that the weather conditions within the model were modified every 30 minutes, rather than daily, in an attempt to demonstrate the concept being presented without providing superfluous data.

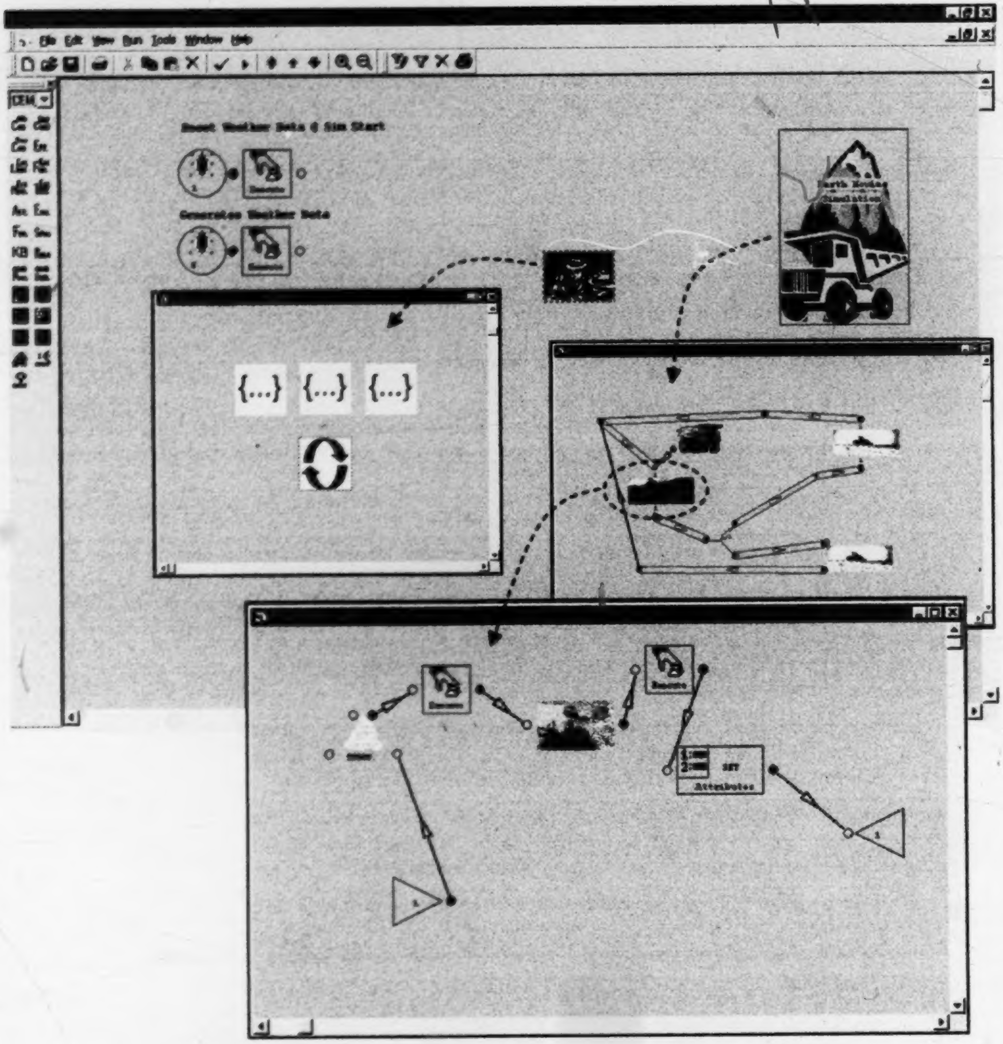

File Edit View Run Tools Window Help
Reset Weather Data & Sim Start
Generates Weather Data
{...} {...} {...}
Earth Moving Simulation
SET
Attributes

Figure 3-19. Agent Embedded in SPS Earth Moving Model

Table 3-4. Agent Influence on Loading Times in the SPS Earth Moving Model

Sim Time (min)	Average Temperature (°C)	Precipitation (mm)	7-Day Cumulative Precipitation (mm)	Shovel Productivity (m3/h)	Loading Time (min)	Productivity Factor
0.0	N/A	N/A	N/A	2400.000	2.125	1.000
11.1	14.7	0.0	13.8	2700.372	1.889	1.125
21.9	14.7	0.0	13.8	2700.372	1.889	1.125
32.7	16.4	0.0	14.6	2726.047	1.871	1.136
43.5	16.4	0.0	14.6	2726.047	1.871	1.136
54.3	16.4	0.0	14.6	2726.047	1.871	1.136
65.1	13.9	12.0	16.5	1052.600	4.845	0.439
78.9	13.9	12.0	16.5	1052.600	4.845	0.439
92.6	12.3	0.7	26.0	2374.916	2.147	0.990
103.7	12.3	0.7	26.0	2374.916	2.147	0.990
114.8	12.3	0.7	26.0	2374.916	2.147	0.990
125.9	9.0	10.3	36.7	779.225	6.545	0.325
141.3	9.0	10.3	36.7	779.225	6.545	0.325

3.7 Chapter Summary

This chapter elaborated on the agent structure developed in the previous chapter through the investigation of the agent's ability to make use of different intelligence forms. While only three forms of intelligence were presented, the point behind the presentation of the new agents was to demonstrate that the agent structure developed allows for varying intelligence faculties.

Three single agent modeling application examples were also provided in this chapter to further demonstrate the abilities of the agent. In the first model the agent was employed to investigate the topology of a tunnel construction simulation model resulting in the modification of the simulation model topology through the addition of another train modeling element in the model. The second example demonstrated the need for a variable intelligence faculty within the agent structure. This example demonstrated the need for a neural network agent to apply the effects of weather conditions for an abutment excavation activity to simulate a bridge construction project. The third example demonstrated two features of the developed agent structure. The first feature demonstrated was the agent structure's mobility, illustrated through its ability to function autonomously and roam from modeling environment to modeling environment. The second feature demonstrated the agent's ability to transpose between problems illustrated though an agent of specific intelligence being employed on two different problems. This ability was demonstrated by using the same neural network weather agent employed to address weather impacts on a bridge abutment activity to address the impacts of weather on a modeled earthmoving operation.

Chapter 4: Intelligent Simulation Agents Framework

4.1 Introduction

The previous chapters introduced the agent structure developed and presented several single agent example applications. This chapter builds upon that work with the introduction of a framework for embedded agent simulation in which several agents are employed in a simulation environment. The framework allows for numerous agents to function as previously described, with the agents absorbed in achieving their agendas, and, more significantly, brings forth an organization for the agents so that the total effect of the agent's influence brings about a global benefit for the model being observed.

The chapter is structured as follows. Section 4.2 presents introductory information related to the development of the agent framework. Included in this section is a brief overview of the Theory of Inventive Problem Solving and how it is beneficial for the framework. Section 4.3 presents a case study of the North Edmonton Sanitary Trunk construction project, which is an example simulation that makes use of the embedded agent simulation modeling framework technique. Section 4.4 explores the benefits and advancements made in simulation modeling from the development of the embedded agent simulation modeling framework. Section 4.5 concludes the chapter.

4.2 Agent Modeling Approach

The agents used in the prior examples had been referred to as artificial superintendents or supervisors placed within a simulation modeling environment. These artificial supervisors were employed and instructed by a modeler through a collection of conditions (observations) and effects (actions) as to what their concerns should be. This allowed the modeler to employ agents within a simulation environment to carry out such roles as minimizing equipment idle time, determining and balancing equipment fleets, manpower resource leveling, and so on. Looking towards more complex examples, it can be seen that there must be some sort of organization among the agents if two or more agents are required for a simulation modeling environment. Previously the agents presented were focused solely on their own agenda. Thus, if two of these self-involved agents were to be placed in a simulation environment it is foreseeable that the agents could work against each other if their agendas were in conflict. This concept can be illustrated with an example: suppose an agent is employed as an asphalt plant dispatcher for a company that supplies asphalt to its own paving projects. This agent wants to ensure that all projects receive asphalt in a timely fashion. Suppose a second agent is employed within the same simulation environment focused on the timely completion of a specific paving project. This project manager agent is going to want all haul trucks leaving the asphalt plant to be dedicated to its project. The dispatching agent will want to ensure that haul trucks are spread among all the projects. Without any collaboration between the two agents, their agendas and actions are going to be in conflict over the control of where the haul trucks go. Therefore, it is the intention of the framework to allow the individual agents to focus on their own agendas while providing a method of administrative guidance so that all the agent efforts follow a common direction.

4.2.1 Overview

The concept for the framework makes use of slave agents focusing on their individual agendas while reporting to a master agent, as presented in Figure 4-1. The focus for the slave agents is on their individual agendas and the small portion of the simulation environment affected by their influence. The focus for the master agent is on the entire simulation environment including all the influences of the slave agents. The role of the master agent is focused on the actions of the slave agents in a common, beneficial direction.

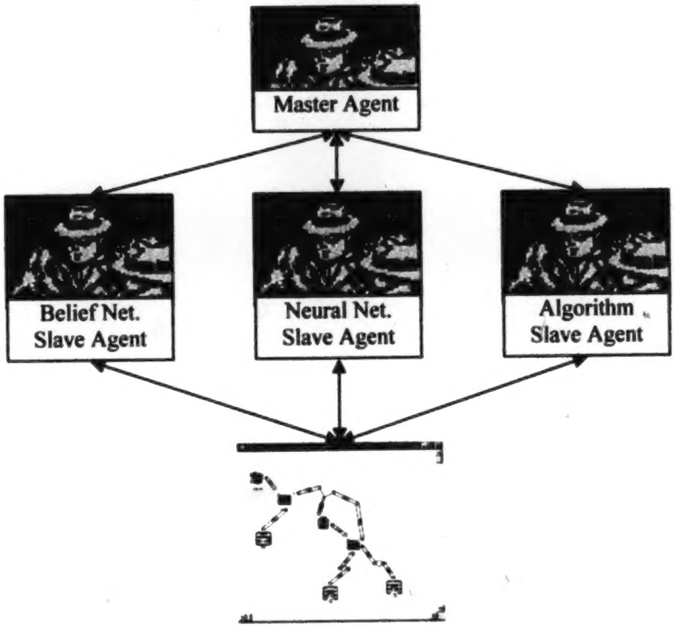


Figure 4-1. Master/slave agent Structure

The fundamental operation of the master agent is no different than that of the slave agents. The master agent is therefore subject to the same agent structure presented in the previous chapters. The master agent is required to make observations, understand these observations, and take action. The major difference between the master and slave agents is the reasoning that is required of the master agent. The master agent is not focused on specific portions of the modeling environment. The master agent assumes an administrative role since it oversees the agents and makes sure that the efforts of slave agents are beneficial to the model as a whole. Due to this administrative role, a different approach is taken for the master agent's intelligence.

The master agent's intelligence must reflect an administrative or management thought process such that the slave agents can be governed. In addition to this, the master agent must be able to analyze the slave agents and the state of the model so that the master agent can construct a comprehensive picture of what is occurring in the modeling environment, thereby formulating and resolving any issues that arise. One analytical tool that has been developed in this capacity is the Theory of Inventive Problem Solving. The Theory of Inventive Problem Solving was developed to provide methods to stimulate thought in problem solvers such that problems with no known solutions can be formulated and solved. It is this problem formulation and solving process that is to be used in the master agent's intelligence facility to allow for issues to be identified and addressed. The second consideration for the master agent's intelligence is to ingrain this problem formulating and solving process with a management or administrative perspective. This will be detailed in the following sections following a brief overview of Theory of Inventive Problem Solving that is relevant to this research.

4.2.2 Inventive Problem Solving

The theory of Inventive problem solving (TRIZ) was pioneered in the mid 1940's through Genrich Altshuller's efforts to understand the evolutionary process of engineered systems as they developed throughout time. Altshuller's interest in the evolutionary development of engineered systems led him to investigate and analyze more than 400,000 patents and derive general problem solving approaches from commonalities found in the documents (Mohammed 2002). In general terms, Altshuller looked at a broad range of documented inventive solutions for problems and identified patterns in the problem solving processes such that tools or methods could be extracted. Between 1946 and 1985 Altshuller developed several problem solving approaches for TRIZ, including ARIZ (Algorithm for Inventive Problem Solving), the 40 Inventive Principles, the Contradiction Table, the Separation Principles, the Substance-Field (Su-Field) Analysis, and the 76 Standard Solutions (Altshuller et al. 1999). These tools did not provide a problem solver with defined solutions for problems, but led the problem solver though approaches that would stimulate thought such that an inventive solution would come to fruition. While each of the tools developed varied in their approach, the common ground Altshuller identified that laid the groundwork for TRIZ is the presence of a contradiction for every inventive problem. That is, for every inventive problem for which no solutions are known, there exists at least one fundamental contradiction underlying the initial problem. The contradiction implies that the desire to improve one of the system's characteristics will result in the undesirable degradation of another part of the system. For example, a football coach wants large heavy players for his team. The coach then wants to make the large heavy players faster and have improved physical fitness, but without sacrificing any of the players' size that would be lost with improved fitness. The contradiction here becomes the players' speed versus their size. The tools Altshuller developed focused on identifying these fundamental contradictions so that they could be identified, formulated, and resolved. Thus, by addressing the contradictions found within the problem, an inventive solution to the problem would materialize that would, in due course, advance the engineered system in its evolutionary life.

The TRIZ tools that are of interest to this research are the 40 Inventive Principles and the Contradiction Table. The following is a brief introduction to these tools. The 40 Inventive Principles is a list of principles that can be used to eliminate technical contradictions. Once a contradiction has been identified and formulated, these principles can be reviewed and applied to the contradiction to stimulate the problem solver to find a solution. The principles are as follows:

1. Segmentation
2. Extraction
3. Local Conditions
4. Asymmetry
5. Consolidation
6. Universality
7. Nesting
8. Anti-weight
9. Prior Counteraction
10. Prior Action
11. Cushion in Advance

12. Equipotentiality

13. Inversion

14. Spheroidality

15. Dynamicity

16. Partial or Excessive Action

17. Shift to a New Dimension

18. Mechanical Vibration

19. Periodic Action

20. Continuity of Useful Action

21. Rushing Through

22. Convert Harm into Benefit

23. Feedback

24. Mediator

25. Self-service

26. Copying

27. Disposable Object

28. Replacement of a Mechanical System

29. Pneumatic or Hydraulic Construction

30. Flexible "Shells" or Thin Films

31. Porous Material

32. Change the Color

33. Homogeneity

34. Rejecting and Regenerating Parts

35. Transform the Physical/Chemical state
36. Phase Transition
37. Thermal Expansion
38. Strengthen Oxidation
39. Inert Environment
40. Composite Materials

The list of principles presents a list of concepts that the problem solver can try to employ to solve their fundamental contradiction. For example, the "Segmentation" principle can be used to divide an object into independent parts, make an object sectional, and/or increase an object's degree of segmentation. Examples of this principle are sectional furniture, modular computer components, and a folding wooden ruler (Altshuller et al. 1999).

While it is advantageous to have a list of 40 inventive principles that can be applied to a problem, the Contradiction Table developed by Altshuller further aids a problem solver by identifying what principles from the list of 40 may be most applicable to a contradiction. Altshuller had defined a contradiction as some circumstance where an improvement to one characteristic of a system results in the degradation of some other component of that system. Keeping this in mind, Altshuller identified 39 systems characteristics that were often associated with technical contradictions (Altshuller et al. 1999). He then was able to present these characteristics in a 39 by 39 matrix. The row heading is the "Feature to Change" and lists all 39 characteristics identified by Altshuller. The column heading is the "Undesired Result" and similarly lists all 39 characteristics. The intersection of the row and column results in a location where the applicable Inventive Principles are listed. The matrix presented in Figure 4-2 contains only the first 5 characteristics identified by Altshuller. The full matrix can be found in Altshuller (1999). From this matrix one should be able to see how the conflicts are represented and the corresponding principles are found. For example, there exists a system in which an engineer wants to improve the length of the moving object and has found that any change to that length creates a contradiction by creating an unfavorable change to the weight of the moving object. In this circumstance, the feature to improve would be the "Length of moving object" and the undesired result would be the "Weight of moving object". Applying this information with the matrix it can be seen that the intersection of "Length of moving object" as a feature to improve and "Weight of moving object" as an undesired result brings forth the numbers 8, 15, 29, and 34. These numbers correspond to the principles Anti-weight, Dynamicity, Pneumatic or Hydraulic Construction, and Rejecting and Regenerating Parts respectively. The engineer could then apply these principles to his contradiction to solve his problem and improve his system. It should be noted that for the cells in Figure 4-2 that are empty, no known solution has been discovered for the contradiction.

			Undesired Result				
			1	2	3	4	5
			Weight of moving object	Weight of non-moving object	Length of moving object	Length of non-moving object	Area of moving object
Feature to Improve	1	Weight of moving object			8,15,29,34		17,29,34,38
	2	Weight of non-moving object				1,10,29,35	
	3	Length of moving object	8,15,29,34				4,15,17
	4	Length of non-moving object		28,29,35,40			
	5	Area of moving object	2,4,17,29		4,14,15,18		

Figure 4-2. Portion of the Conflict Matrix.

4.2.3 Agent Framework Development

The agent framework provides a means for a team of agents to be employed within a simulation environment and, under a common focus, evolve a simulation from a crude initial representation of a system to that of a refined model. In developing this framework, a master/slave agent structure has been adopted providing a governing influence over the agent actions within the environment. It has been recognized that while the role of the master agents does not place any burdens on the constructs of the agent structure developed, it does have implications on the reasoning carried out by the master agent. The master agent is required to, in a managerial capacity, take in the entire modeling environment and identify, formulate, and resolve issues regarding the positive influence of the overall actions of the agents on the long term improvement of the model being observed. The tool suggested to provide the master agent with this reasoning capacity is the TRIZ Conflict Matrix.

The Conflict Matrix provides a problem solver with a means of relating possible methods to achieve a solution back to the initial problem through the identification of what is to be improved (Feature to Improve) from the problem and what negative effects (Undesired Result) come from that desired improvement. While the specific Conflict Matrix developed by Altshuller is applicable for the most part to the evolution of engineered systems and is not created for the identification, formulation, and resolution of administrative affairs, the concept of the matrix and how the problems are identified and related to principles that can lead to solutions can be abstracted to develop the master agent's reasoning.

The abstraction of the concept of the Conflict Matrix for the master agent's reasoning for construction engineering and management (CEM) issues required a similar development process to the original construction of the Conflict Matrix. In the development of the Conflict Matrix Altshuller (1999) investigated the evolution of engineered systems and was able to identify improvement and undesirable features in problems which described the contradictions. Altshuller then presented these features in very basic, non-descript way so that these features could be applicable to contradictions found within any engineered system. Following a similar investigation for the development of a CEM Conflict Matrix, the definition of project management is examined. Ahuja, Dozzi, and AbouRizk (1994) define project management as "the art and science of directing human and material resources to achieve stated objectives within the constraints of time, budget, and quality and to the satisfaction of everyone involved." From this definition the terms "resources", "time", "budget", and "quality" are identified as fundamental features for management issues.

Another aid that can help define fundamental features of project management is the project management model shown in Figure 4-3 (Ahuja et al., 1994). Figure 4-3 presents a project management model as a three-dimensional matrix which illustrates the relationships between management functions, processes, and stages. The figure gives light to what is managed during a project (Functions), the methods used to manage these items (Processes), and the times during a project when those items are managed (Stages). The relevance of this diagram to the development of a CEM Conflict Matrix is that the functions further identify what is managed during a construction project. This figure further identifies "Scope", "Communications", "Human Resources", and "Risk" as CEM issues. Therefore, from the investigation of these two items, the terms resources, time, budget, quality, scope, communications, human resources, and risk have been identified as generic CEM terms that describe fundamental CEM issues. One further term that is not identified in the project management definition or the project management model is safety, and it will be added to the list of CEM features.

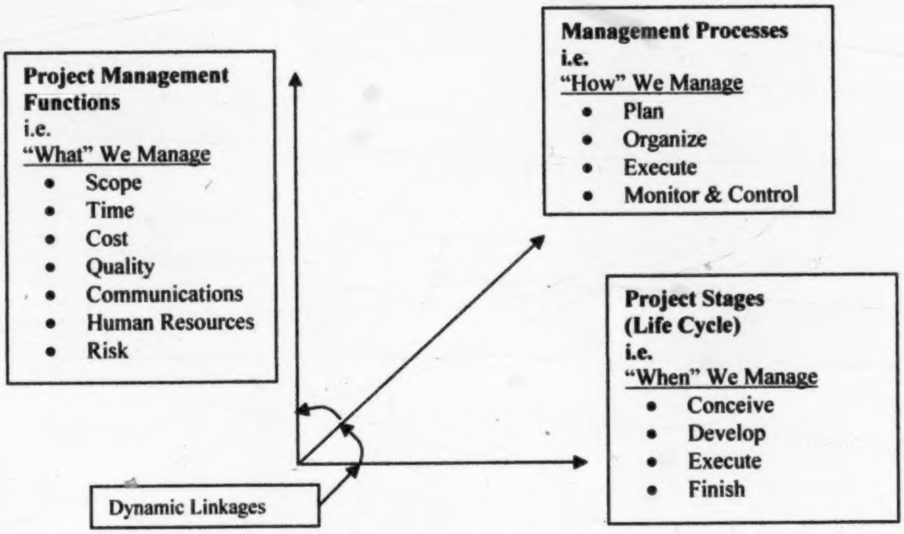

Project Management Functions
i.e.
"What" We Manage
Scope
Time
Cost
Quality
Communications
Human Resources
Risk
Management Processes
i.e.
"How" We Manage
Plan
Organize
Execute
Monitor & Control
Project Stages
(Life Cycle)
i.e.
"When" We Manage
Conceive
Develop
Execute
Finish
Dynamic Linkages

Figure 4-3. Project Management Model (Source Ahuja et al. 1994)

The list of CEM features arrived at can be used to construct the beginnings of a CEM Conflict Matrix. The CEM features can form the "Features to Improve" and "Undesirable Result" variables for the matrix. The resulting matrix is presented in Figure 4-4. This matrix will allow the master agent to recognize issues with the simulation environment in terms of the fundamental CEM features shown. The next step in developing this matrix is to define the list of principles that will fill the matrix and resolve the CEM contradictions.

		Undesirable Result							
		Scope	Time	Cost	Quality	Commu	Resour	Safety	Risk
Feature To Improve	Scope								
	Time								
	Cost								
	Quality								
	Communications								
	Resources								
	Safety								
	Risk								

Figure 4-4. CEM Conflict Matrix

The matrix presented in Figure 4-4 illustrates a means by which the master agent can conceptualize contradictions for CEM systems. The second aspect of the master agent's reasoning facility indicates that the master agent's reasoning should also reflect management or administrative characteristics. These management characteristics can be ingrained in the principles within the matrix. A management policy or style can be embedded within the matrix by breaking down the management policy into its fundamental principles and indicating where those principles can be applied in the matrix to provide solutions for the various CEM contradictions. The matrix and policies can then be presented to the master agent in the form of a belief network, as shown in Figure 4-5. A contradiction in the belief network can be represented by setting the states of the belief nodes that represent the CEM features to true and then evaluating the network to identify the corresponding management principles that would assume significant beliefs of true. One concern when doing this is that there are numerous approaches to management resulting in numerous management styles. It is therefore not possible to construct a CEM Conflict Matrix that could fundamentally represent all management styles. Thus, specific CEM Conflict Matrices belief networks would have to be created for each style of management so that the master agent can make use of the CEM Conflict Matrix belief network relevant to the style of management desired for the modeling environment.

The resulting CEM Conflict Matrix provides a nonspecific reasoning mechanism that allows the master agent to identify, formulate, and resolve issues found within the simulation environment. The final development concern for the agent framework and master agent's reasoning is the transfer of information to and from the master agent. The master agent's reasoning has been developed using general CEM features. The slave agents have been developed in such a manner that they are specific to the issues that they deal with. The master agent therefore requires the specific information it receives from its slaves to be translated into non-specific terms. Similarly, the master agent must translate its non-specific actions into specific actions that can be applied to the modeling environment. This process is illustrated in Figure 4-6. The translation can occur at two locations within the modeling environment. The slave agents can be instructed in their child elements to translate the information sent to and received from the master agent. Conversely the master agent can be instructed though its child elements on how to translate the information it receives and sends.

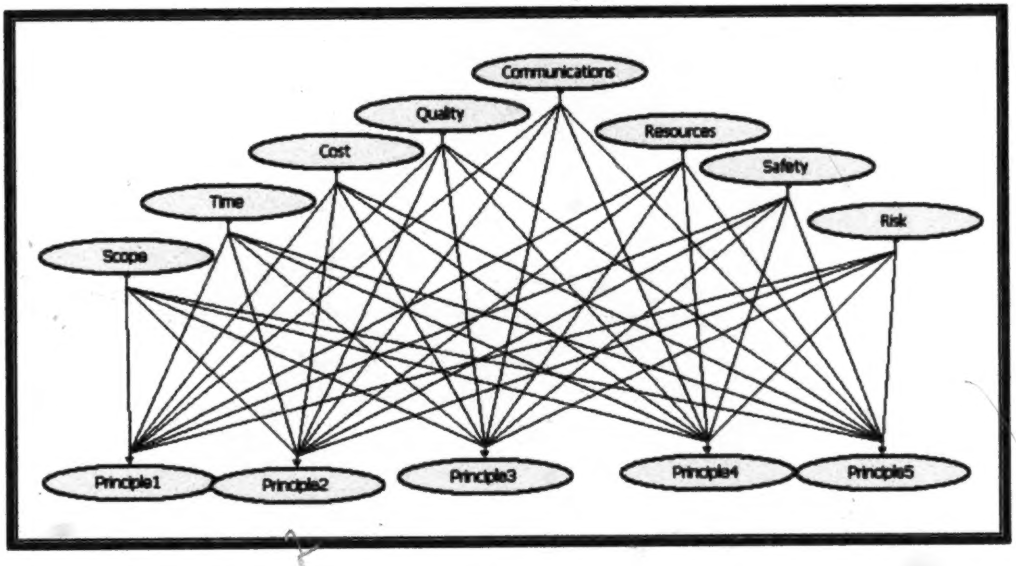

Communications
Quality
Resources
Cost
Safety
Time
Risk
Scope
Principle1
Principle2
Principle3
Principle4
Principle5

Figure 4-5. CEM Conflict Matrix in Belief Network Representation

The resulting agent framework developed is presented in Figure 4-7. The frameworks allows for a variety of slave agents to carry out their personal agenda while reporting information or actions to a master agent. The diagram presented in Figure 4-7 is broken down into six components. This is done to exemplify the process or significance of each portion of the framework.

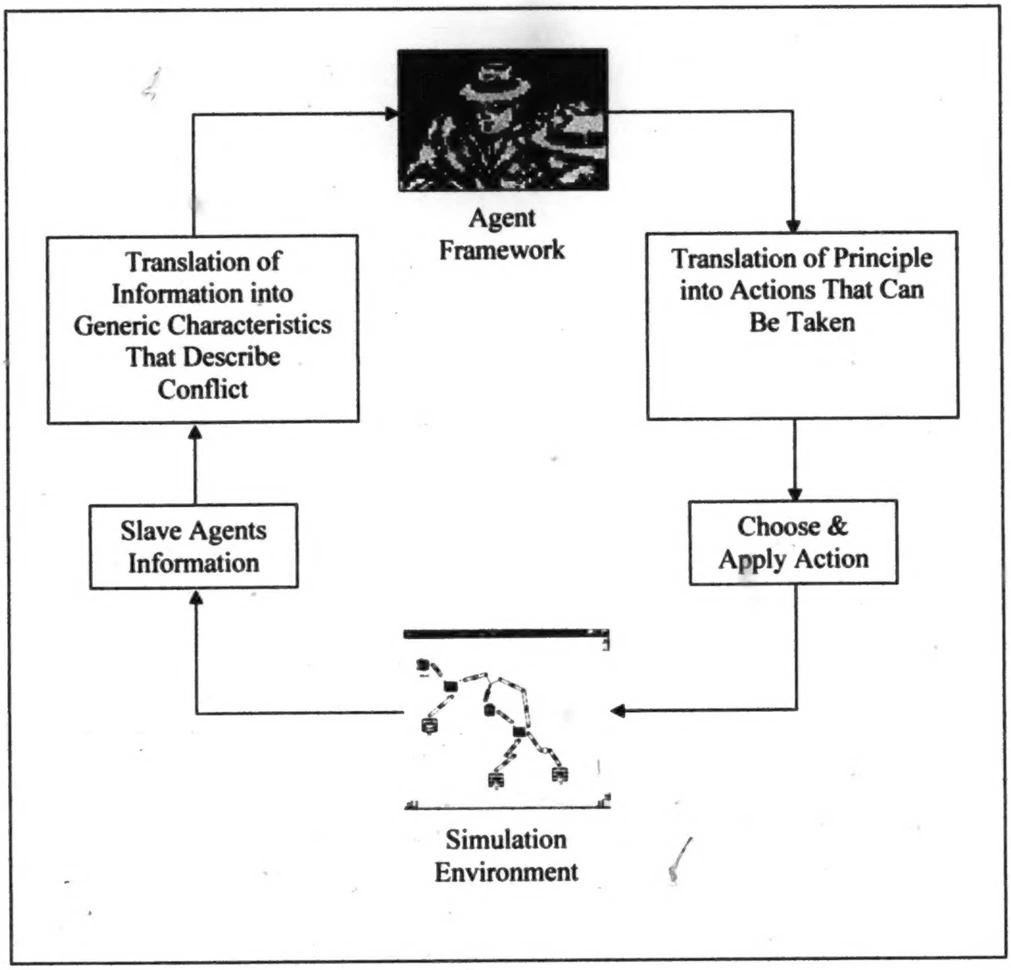

Agent Framework
Translation of Information into Generic Characteristics That Describe Conflict
Translation of Principle into Actions That Can Be Taken
Slave Agents Information
Choose & Apply Action
Simulation Environment

Figure 4-6. TRIZ Abstraction for Master Agent's Reasoning

The first component has been identified as the *Model Representation*, which is where a system is modeled and all the advantages of simulation have been captured for the benefit of the overall problem-solving process. The problems are conceptualized through representation of the system in the *Model Representation* component. It is also where simulation is used to provide precision with respect to the issues of complexity, uncertainty, and variability found in the model.

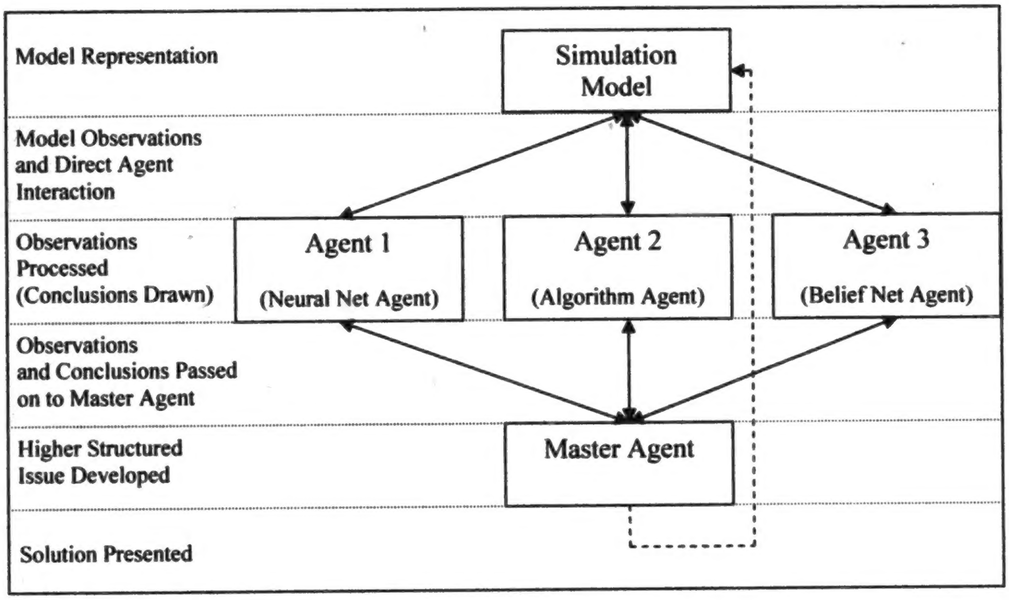

Model Representation
Simulation Model
Model Observations and Direct Agent Interaction
Observations Processed (Conclusions Drawn)
Agent 1
(Neural Net Agent)
Agent 2
(Algorithm Agent)
Agent 3
(Belief Net Agent)
Observations and Conclusions Passed on to Master Agent
Higher Structured Issue Developed
Master Agent
Solution Presented

Figure 4-7. Embedded Agent Framework

The *Model Observations And Direct Agent Interaction* component is the part of the problem solving process where the individual slave agents make their observations of the model and attend to their respective agendas. The agents are able to take action on the model at this level if their agendas deem that they are required to do so. From the diagram it can be seen that this component has a two-way interaction between the model and the slave agents. The slave agents are taking information from the model and acting upon the model when required.

The *Observations Processed* component is where the agents process the information from the observations made of the simulation model. One can see that there is some overlap between this component and the previous one, as the model observations must be processed by the slave agents before they can individually act upon the model.

The *Observations and Conclusions Passed on to Master Agent* component is the portion of the framework where the slave agents report to the master agent the observations, inferences, conclusions, and actions they have made. The master agent then takes in this information and begins to formulate an overall global perspective of the simulation model in the *Higher Structured Issue Developed* component.

The act of having the slave agents report to the master agent focuses all the agents' actions towards the overall development of the simulation model from its initial beginnings to a finished product. While the individual agents may have their own agendas to address specific issues within the model, effort must be made to ensure that by solving these smaller problems individually a greater good is achieved for the entire model. The crucial process for this endeavor is the master agent receiving information from its slave agents to gain an overall idea of what is occurring in the model, allowing the master agent to processes this information with respect to its agenda.

4.3 Case Study

The following case study of the North Edmonton Sanitary Trunk (NEST) tunnel construction is provided to demonstrate the developed agent structure. This project was carried out making use of lean construction management principles. The agent framework is embedded into the simulation environment to allow for several slave agents to address individual issues found within the construction project. The master agent is employed in the simulation environment to ensure that the slave agents carry out their responsibilities while following the Lean Construction philosophy that governed the execution of the project. While the information provided in the following sections details the use of the agent framework in the modeling environment, additional information can be found regarding this study in Appendix 8.

4.3.1 Introduction

For this project, Lean Construction Principles were applied as a method of management. The intention behind applying these principles to the project was to streamline tunnel construction through the identification and elimination of waste. The intention for this example is to duplicate some aspects of the actual construction management processes through the use of the embedded agent modeling framework. This example will then show that these agents are capable of assuming roles within a simulation environment and carrying out significant tasks that previously could not have been done by anyone but the modeler.

The NEST tunnel is a 1600m-long, 2.9m finished diameter, segmentally lined tunnel. In this example two key issues will be explored in the simulated model. The first of these issues is the addition of a second train for the spoil removal from the tunnel boring machine (TBM). The second issue deals with the scheduled completion of the project. In reality, this tunneling project was only one activity of a much larger project, which was looking at the growing needs of the expanding residential community in the area. Therefore the tunneling project had an imposed completion date of April 1, 2002 where the tunnel was to be completed and commissionable. In addition to this, a start constraint was imposed on the tunneling activity since the TBM and its crew were employed on another project and unable to start actual construction until July 23, 2001. This obviously left a distinct window for tunnel construction. Another constraint that further constricted this window of construction was the desire to keep the local residents in the area content with a minimum number of disruptions from the construction. Tunneling can be carried out as a twenty four hour operation by making use of day and night shift crews. Since the location of the tunnel was in a residential area, the idea of a night shift for this project was unfavorable as the noise from construction operations during the night would bother the residents. It was therefore decided that construction was to be carried out as a single day shift operation until the second shift was absolutely required to ensure meeting the completion milestone in a timely fashion.

4.3.2 Lean Construction Concepts

Lean thinking was pioneered in the 1950's by Taiichi Ohno, who was the Toyota executive who identified seven types of muda (Womack and Jones 1996). *Muda* is the Japanese word for waste, and, in Ohno's lean thinking views, *muda* was any human activity that absorbed resources but created no value. The seven types of *muda* identified by Ohno were defects in products, overproduction of goods not needed, inventories of goods awaiting further processing or consumption, unnecessary processing, unnecessary movement of people, unnecessary transport of goods, and waiting by employees for process equipment to finish or for an upstream activity to complete. By identifying these seven types of *muda*, Ohno was able to develop "lean thinking" principles that focused on the elimination of these waste items from his manufacturing processes.

One of the key components to lean thinking is value. Ultimately the customer determines what value is. Value for one customer can be a high performance product. For another customer the value may be found in the safety and quality of the product. Lean thinking is a management philosophy that identifies value and tries to achieve that value as efficiently as possible, i.e. without waste. This line of thought does not necessarily imply carrying out cost-cutting endeavors that result in job elimination, diversion of revenues, and other harsh measures, but looks in a more positive perspective towards streamlining processes. It tries to do more and more in terms of value with less and less in terms of waste. Thus the philosophy tries to do more with less human effort, less equipment, less space, less time, less quality defects, and so on.

Clearly this approach to streamlining a manufacturing process can be viewed as beneficial to other areas of industry. One group that has identified the benefits of Lean Thinking is the International Group for Lean Construction (1999). A prime example of the International Group for Lean Construction can be observed in Tommelein (1998), which presents a lean construction technique called pull-driven scheduling and applies that technique to a material management simulation for a fast-tracked process plant construction project. When presenting the lean construction technique, Tommelein presented five lean thinking production techniques that were applicable to the lean construction scheduling techniques. These policies are (Tommelein 1998):

1. Stopping the assembly line immediately to repair defects
2. Pulling materials through the production system to meet specific customer demands
3. Reducing overall process cycle time by minimizing each machines change-over time
4. Synchronizing and physically aligning all steps in the production process
5. Clearly documenting, updating, and constantly reporting the status of all process flows to all involved parties

It is these five policies that were used when managing the actual construction of the NEST tunnel. Therefore these same policies should be used when constructing the CEM Conflict Matrix. The resulting matrix, presented in Figure 4-8, was then transposed to the belief network presented in Figure 4-5.

	Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk
Scope	▨	2,3,4	3,4	1	1,5	1,4	1,5	1
Time	2,3,4	▨	2,3,4	1,2,3,4	2,3,4,5	2,3,4	2,3,4,5	2,3,4
Cost	3,4	2,3,4	▨	1,3,4	3,4,5	3,4	3,4,5	3,4
Quality	1	1,2,3,4	1,3,4	▨	1,5	1,4	1,5	1
Communications	1,5	2,3,4,5	3,4,5	1,5	▨	4,5	5	5
Resources	1,4	2,3,4	3,4	1,4	4,5	▨	4,5	4
Safety	1,5	2,3,4,5	3,4,5	1,5	5	4,5	▨	5
Risk	1	2,3,4	3,4	1	5	4	5	▨

Figure 4-8. CEM Conflict Matrix with Lean Construction Principles

4.3.3 Tunnel Construction Example

Figure 4-9 is a screen capture of the modeling environment in which the tunnel construction model has been placed along with a collection of agents. The tunnel construction model placed in the simulation environment happens to be the actual NEST tunnel that was used to carry out the pre-construction analysis for the NEST project. Therefore the agents will be carrying out actual tasks that were previously done by the modelers. The agents placed in the modeling environment have followed the master/slave framework described. For this example there is one master agent overlooking four slave agents. The master agent has been provided with the intelligence described in the previous section and shown in Figures 4-5 and 4-8. The four slave agents have been titled Production Agent, Schedule Agent, Soil Agent, and Operations Agent. The Production Agent is a rule-based agent. This agent works with the Soil Agent to determine the mathematical distribution for the TBM's advance rate based on expected soil conditions and number of labor shifts (day and night) for the upcoming simulated day. During a simulation run, the Productivity Agent is responsible for directly updating the mathematical distribution for the TBM's advance rate so as to keep the simulated excavation progress as close to reality as possible. The Productivity Agent also provides the current productivity distribution to the master agent as well as the other agents in order to keep everyone informed and aware. The Schedule Agent is another heuristic rule-based agent that has been given a rule base to analyze scheduling concerns in the model. Basically the Schedule Agent is looking at the current schedule and forecasting to determine if a night shift should be suggested. The Schedule Agent does not act upon the model, but instead reports its analysis to the master agent. The Soil Agent uses a soil prediction algorithm and a borehole log database for its intelligence to allow it to predict soil conditions that are to be expected during construction. Similar to the Schedule Agent, the Soil Agent does not act upon the model and only offers information to the other agents. Lastly the Operations Agent is a belief network agent that uses its intelligence to analyze the efficiency of the construction operations and help determines when a second train is required. This agent does not act on the model, but instead reports its findings to the master agent. Figure 4-10 gives an illustrated summary of the agents' interaction under the framework.

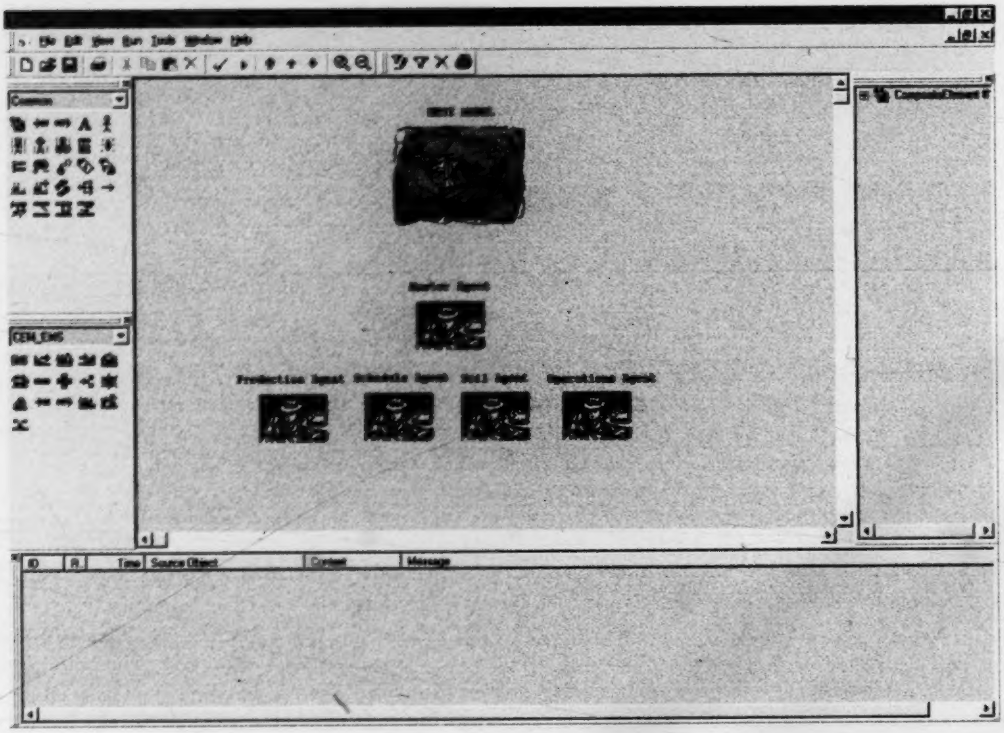

File Edit View Run Tools Window Help
Master Agent
Production Agent
Schedule Agent
Soil Agent
Operations Agent
ID
R
Time
Source Object
Content
Message

Figure 4-9. Agent Embedded Model for NEST Tunnel Construction

From Figures 4-9 and 4-10 it can be seen that the master agent is supervising four slave agents who are individually analyzing specific portions of the model and reporting conclusions. It now becomes the master agent's responsibility to digest the information received from the slaves and any direct model observations made by the master agent. The master agent then processes this information to determine if any conflicts have transpired that can be solved by lean construction concepts.

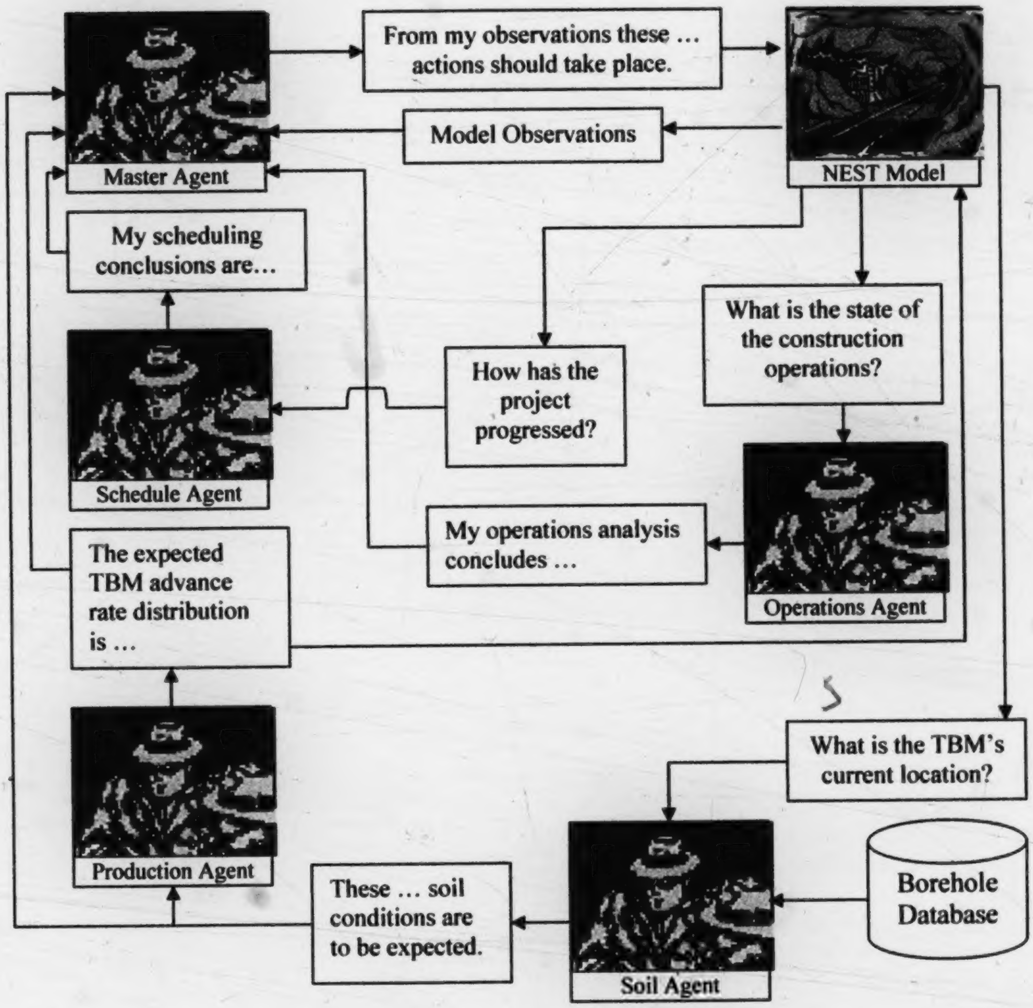

From my observations these … actions should take place.
Model Observations
Master Agent
NEST Model
My scheduling conclusions are…
What is the state of the construction operations?
How has the project progressed?
Schedule Agent
My operations analysis concludes …
Operations Agent
The expected TBM advance rate distribution is …
What is the TBM's current location?
Borehole Database
Production Agent
These … soil conditions are to be expected.
Soil Agent

Figure 4-10. Agent And Model Interaction Flowchart

In terms of specifics for this example, the focus is directly on two problems: when should a night shift should occur and when a second train should be added to the operation. The master agent has therefore been set up to process the information from the model and slave agents so that if there appears to be a problem with the operations (i.e. trains) this will translate into a resources vs. cost conflict, and if there is a scheduling problem (night shift needed), this will translate to a time versus cost conflict. From Figure 4-8 it can be seen that a resources vs. cost conflict can be solved through lean principles 3 and 4 and that the time versus cost conflict can be solved by lean principles 2 through 4. This results in three possible principles that can be applied to the model based on the conflicts presented. These principles are (Tommelein 1998):

- Principle 2 - "Pull material through the production system to meet specific customer demands"
- Principle 3 - "Reducing overall process cycle time by minimizing each machine's change over time"
- Principle 4 - "Synchronizing and physically aligning all steps in the production process".

To simplify the example it is assumed that since the tunnel construction follows a linear construction method it has been well practiced, refined, and synchronized suggesting that lean principle 4 should not be taken into consideration. This deletion leaves principles 2 and 3 to be of concern for the master agent. Principle 2 indicates pulling material through the system to meet customer demand. In this system, we can translate the customer demand to be that of the TBM "demanding" to be utilized as much as possible in order to meet its completion milestone. Therefore, if principle 2 should arise, the master agent will look at the tunnel construction process and see where improvement to the construction process can be made to improve the TBM advance rate. Therefore, for this example, if principle 2 arises the master agent should be instructed to first look at the model to see how many trains are in the system. If there is only one train in the system, then a second one should be added. If two trains are in the system, then the agent should consider adding a night shift.

If the master agent should happen to come across principle 3 as a solution for one of its conflicts, then the master agent should act accordingly. The third principle looks at minimizing the changeover time between pieces of equipment. For the given problem there is only one instance where this can be addressed. This is the changeover time for the muck train leaving the TBM and returning. This situation can be rectified through the addition of a second train. Therefore, if the master agent should happen to come upon principle 3, it should be instructed to count the number of trains in the system. If there is only one, then a second train should be added.

In reality the actual construction of the NEST Tunnel began on July 23, 2001. A critical date was determined for the addition of the second train, but was not used since operations began using two trains to support the TBM. The second shift started on October 1, 2001. Project completion occurred on February 8, 2002. Starting the simulation model on July 23, 2001, the agents had determined that the second train should have been added on August 24, 2001 at 265 m, and the second shift should have started on March 13, 2002, for a simulated completion date of April 5, 2002. While the simulated dates do not reflect the actual dates from construction, it should be noted that if the construction would have progressed as directed by the simulation dates, the project would have been able to provide for a timely, cost-effective completion while minimizing the disturbances to the residents of the area. The actual construction provided for an estimated 94 night shifts. It should be noted that this number does not account for holidays, down time, and other reasons that a night shift would not be used. The simulated construction provided for 24 night shifts which is a considerable improvement.

The initial study of the NEST tunnel was carried out in March 2001 (Farrar et al. 2001). For the study, each tunnel construction issue was analyzed through a compilation of simulation model runs. The problem solving approach taken for these individual issues was an educated guess and test process where the modeler carried out the iterative process of changing the model, observing the simulation results, and repeating the process until acceptable results were obtained. Once the individual solutions were found, the results were pieced together creating a plan for tunnel construction. The process took approximately two to three weeks to complete with three simulation modelers. The construction execution plan developed from the simulation study was based on a March 30, 2001 project start and anticipated a need for a second train at the project start with the second shift starting on August 1, 2001 (100 meters into the tunnel construction). This plan brought about an April 4, 2002 completion.

In comparison to the agent embedded simulation modeling approach, the agent framework problem solving approach took approximately 12 hours to complete: two hours were required to build the model and 10 hours were required in simulation time. While the start date differed between the two models a difference can be noticed in the determination of when to employ a second shift on the tunneling project. The previous study made use of a March 30, 2001 start date and employed the second shift on August 1, 2001. This plan resulted in 247 night shifts. The agent framework made use of a July 23, 2001 start date and determined that the second shift would be required on March 13, 2002 resulting 24 night shifts. One significant factor that could have contributed for this difference is the agents' ability to take into account large amounts of data through their artificial intelligence capabilities and make decisions in simulation real time there by reacting to their dynamic environment and influencing its direction. In contrast to the approach taken in the initial study where the model was somewhat static and introduced the second shift or second train as directed by the modeler and his knowledge of the issue, the agent framework was able to access, and process significant amounts of borehole logs, scheduling constraints, operations information, and productivity information and make decisions that would influence the model in terms of determining when the second shift and train should be added. Comparing the two problem solving approaches to what actually happened during construction, the agent embedded framework approach shows more reliable results for plan development over the traditional simulation evaluation method. Furthermore, with respect to the introduction of the second shift to the project, the traditional evaluation method produced results that were conservative in comparison to reality, while the agent embedded framework demonstrated the decision made in reality was conservative.

Lastly is should be noted that the lean construction principles applied to this problem were used to demonstrate the concept of the master agent and the agent framework. In general, the lean construction concepts demonstrated for this problem that there is a lower cost approach to the tunnels construction. However, for this problem, the lower cost approach is not of concern for the City of Edmonton since their main concern was time. Thus master agents' management policies for this example could be better represented with a management strategy other than lean construction.

4.4 Benefits of Agent Embedded Modeling

The overall contribution of this work is the establishment of a new problem-solving framework to address CEM issues. This work has created intelligent objects that can be placed within a simulation modeling environment and relied upon to carry out tasks, assume responsibilities, provide intelligent insight, and solve problems. Furthermore, this research presents a framework for embedding agents within a simulation modeling environment that allows a variety of agents to be employed to address many different issues on many different levels while maintaining a degree of organization and focus.

In addition to these major contributions, there are many other tangibles to be realized. First, this research created a new tool for simulation modeling. This tool is the structure for a generic agent that is able to observe, think, and act in simulation real time. This tool has created the ability for a modeler to capture intelligence and employ it in an automated fashion to carry out tasks. No longer is the modeler required to be a construction and/or simulation expert, and no longer is the modeler required to invest large amounts of time in iteratively exploring a model. Another tangible is the developed agent's ability to evaluate and act upon the topology of the model. This ability to act upon the model's topology surpasses any previous agent abilities that can be found in the available literature. To date there have been two types of agents developed for simulation modeling. The first group of agent applications are those used in distributed simulation modeling. These agents act as model ambassadors and generally communicate information between simulation models or environments to coordinate between distributed models. The second group of agent applications in simulation are either used to add detail to a model, such as an intelligent solider in a wargame simulation, or the agent is looking at improving the model only through modeling element parameter evaluation. The new generation of agent structure developed through this work has the ability to assume all the roles mentioned above, in addition to looking at the topology of the model and changing it by adding, removing, or modifying modeling elements within the modeling environment.

4.5 Chapter Summary

This chapter presented the beginnings of a new problem solving technique focused on addressing issues found within the CEM community. The problem solving method presented is carried out through an embedded agent simulation modeling framework technique. The framework makes use of embeddable intelligent objects in the simulation modeling environment that act as artificial observers for the processes occurring in the model.

The agent framework presented is founded upon a master/slave agent relationship. The relationship provides a method for employing a team of slave agents in a simulation environment under the guidance of a master agent. Under this framework the slave agents are allowed to attend to their individual agendas and concentrate on specific areas of a simulation model, but were required to report their information and actions to a master agent. This reporting procedure was designed to ensure that the team of agents had an overall guiding influence to ensure that the team of agents and their actions had a positive effect on the simulation environment as a whole.

Chapter 5: Discussion

5.1 Research Summary

The initial goal of this research was to develop a new method for approaching construction engineering and management (CEM) related issues. The approach taken combined the benefits and precision of simulation modeling with the inclusion of intelligent aids called agents. Simulation modeling has proven itself a viable tool within the CEM community through its ability to capture the randomness and variable nature of construction in a proficient manner. The inclusion of intelligent agents in the simulation modeling environment offers greater modeling and problem solving capabilities to people of any modeling and construction management skill level. The agents aid the skilled simulation modeler with access to construction experience and knowledge and offer a similar service for the experienced construction manager through access to simulation modeling information and capabilities. Furthermore, the agents provide aid to both groups within their domain by being able to capture their knowledge and experience and assume roles within the environment. The resulting approach provides an autonomous simulation environment in which the modeler constructs an initial model and then employs the agents through their framework to refine the model into a finished product.

The focus of the research was developing the generic agent structure and agent framework so that various types of intelligence facilities could be applied on any simulation model. In doing this, five types of agents were presented. Four of these agents were developed as slave agents that could make use of belief networks, artificial neural networks, algorithms, and rule-based reasoning facilities. The fifth agent presented is the master agent. This agent is able to govern the slave agent actions through the use of intelligence founded on Theory of Inventive Problem Solving.

The structure for the agents consists of three components: the parent agent element, the condition element, and the effect element. The parent agent element is the engine for the agent. It is the intelligent or reasoning component of the agent. The other two components form the agent's agenda. The agenda is where the modeler instructs the agent as to its role within the simulation modeling environment. The condition element allows the modeler to instruct the agent as to what the agent is to observe in the modeling environment during the simulation run. The effect element provides a similar service by letting the modeler outline to the agent the possible actions it can carry out.

The agent framework is developed making use of a master/slave agent structure. The slave agents role within the framework are to focus on small, specialized portions of the simulation modeling environment to address concerns found there. The slave agents are also required to report information and actions to their master agent. The master agent is responsible for ensuring that all of the slave agents work towards benefiting the entire simulation model. This supervision is required to prevent the individual slave agents, acting with good intentions to improve their portion of the simulation environment, from causing detrimental effects on other areas of the simulation model.

5.2 Evaluation of Research Objectives

For this research one major objective was identified as well as three sub objectives. The first objective was to crate a framework for embedding agents in simulation models for construction management issues with respect to the iterative simulation model evaluation process identified in Figures 1.1 and 1.2. In doing this, the first sub-objective identified was the development of a mechanism for the agent so that it could be applied to many different models without complication. Furthermore, the second identified sub-objective addressed agent knowledge encapsulation in a similar manner by developing an agent mechanism that allowed the agents intelligence tools to be applied to many different models without complication. The last sub-objective looked at the development of an agent mechanism that could observe and act upon a model in terms of the details within the simulation model as well as the topology of the model.

In evaluation of the major objective for this research an agent embedded simulation modeling framework was developed in which individual agents could be placed within a simulation modeling environment and used to provide knowledge and effort to solve problems for a modeler. The detail to which this framework was developed allows for a modeler to simulate a decision making processes using a team of agents that can encompass more information than capable by a human person. This is a definite contribution to science and indicates that something of value was obtained from the development of the framework. Furthermore, if one were to compare the agent framework to computer programs that are able to perform similar duties, a notable difference between the two methods is the agent frameworks dynamic characteristics. While there are other methods of attaining the same problems solving process as presented by the framework, the other processes are not as transportable as the agent framework process. The agent framework places the decision making process and the information and intelligence to carry it out directly in the modeling environment. As was demonstrated through the examples provided, this framework can be transported from one problem to the next with little complication and effort. Comparing the framework to previous research efforts such as those by Van Tol (2000) and McCabe (1997), the prior problem solving methods developed were static with respect to how they solved problems. These research efforts were programs that ran a simulation model, obtained information from the model, changed the simulation model, and ran the simulation model again in an iterative problem solving process. While these methods carried out a similar evaluation process, they did not react and adjust to changes in the simulation model, which the agents are able to. These programs simply processed information and did not try to influence the modeling environment. Any change in state or condition of the simulation model went unnoticed by these programs. To further demonstrate the dynamic versus static comparison between the agent framework and prior research efforts, the agent framework is in a state of constant observation of the model which allows the agents to continually update their evaluation criteria and how they influence the model. Due to the static nature of computer programs, the prior research developments do not carry this ability simply because they are designed to process information in a structured manner.

In evaluation of the progress made in this research for the three identified sub-objectives, a generic agent structure and agent framework has been developed that appears to be common in its application. The agent structure was developed following a very basic process. The agent is designed to make observations, process that information using some sort of intelligence, and then take action based on conclusions drawn from the processed information. This ability has been demonstrated through the examples provided in this thesis. Thus, there is a high level of confidence when stating that the agent structure developed can function satisfactorily within any environment. The agent structure was further developed to support the second sub-objective requiring the agent being able to make use of its intelligence and knowledge in many different simulation models. The agents presented in this research show that the type of intelligence used by the agent can vary. One agent can use belief networks for its intelligence and another agent can use algorithms for its intelligence. Both agent have the same basic structure and mechanics. The intelligence is just an interchangeable feature of the agent structure. Effort was also taken to demonstrate that the knowledge encapsulation can be done in such a manner that the agents can be applied to different models while making use of the exact same intelligence. This was demonstrated through the two excavation simulation examples, where an agent with a weather productivity artificial neural network was able to apply the impacts of weather to distinctly different models. The bridge abutment excavation model was a model of a CPM network and the earthmoving model was an operations process model of an earthmoving activity. Essentially the agent was copied from one model to the next and functioned appropriately with little modification to the agent. Lastly, with respect to the third sub-objective the examples presented in the research demonstrated that the agent developed is able to influence the modeling environment through the modification of modeling element parameters and topology of the model. The agent can change the models topology through the addition, removal, or modification of modeling entities, elements, and parameters.

In summary of the research objectives evaluation, it is felt that all objectives were achieved. Further work could be done to develop more domains of intelligence for the agents so that they can approach problems that are not of a similar nature as presented in this thesis.

5.3 Limitations

One limitation of significant concern regarding this research is an over reliance on the intelligence found within the agents. Misuse of the embedded agent simulation modeling framework is foreseeable if the modeler becomes over reliant on the agents' intelligence without any insight into its makeup or how that intelligence is to be used. Thus, the requirement for possessing significant simulation and construction knowledge has been replaced with a different requirement of understanding the reasoning and role of the agents. This concern is raised even after examples were presented to demonstrate the agent structure developed and intelligence used by the agent can be directly transposed from modeling environment to modeling environment.

A second limitation of the developed framework is the modeling environment in which it was developed. The simulation modeling environment offered many advantages for research and development with features such as hierarchical modeling and linking modeling element parameters, but the manner in which the simulation program currently handles reference to artificial intelligence programs outside of the simulation program heavily taxes the performance of the computer. The current state of the simulation program will not offer embedded agent simulation modeling framework capabilities much greater than those presented in the NEST tunnel construction case study. However, this concern should be resolved with the next generation of development of the simulation program.

5.4 Summary of Research Contributions

The described research has brought forth numerous contributions that will change the perspective on simulation modeling and how it is used within the CEM community. Paramount to these contributions is the development of the agent embedded simulation modeling approach for addressing CEM issues. In accomplishing this feat, other contributions can be identified, beginning with the development of the agent structure at the foundation of this research.

The development of the agent structure provides a simulation modeler with a generic method in which intelligence can be captured and utilized in a simulation modeling environment through an automated model refinement process. Prior to this research, the literature review had identified two main branches of agent use in simulation. One branch identified simulation agents as a means of managing distributed simulation modeling systems. The other branch made use of agents as intelligent components of a simulation model that provided further realism to the abstracted modeled system. The agent developed under this research has been done following the intent of the second branch but with extended flexibility in regards to its role definition. The agent's role is defined by the modeler employing the agent in the simulation environment through the agent's agenda. Thus the agent developed under this research does have the capacity to assume an intelligent role in the simulation model, but has been developed primarily with respect to play an intelligent observer role within the simulation modeling environment, capable of analyzing the environment and influencing the development of that environment with respect to the instructions provided by the modeler. The desired end result of the agent development is the creation of intelligent simulation modeling environments in which the models are able to analyze and modify themselves. In developing this capacity for model modification, an agent has been developed with a capability not currently found in the literature. This capability is the agent's ability to alter the model's topology through the addition and removal of modeling elements in the simulation environment. Prior to this development, agents that were created for improvement considerations to models only addressed modeling element parameters. Furthermore, the *Observation Assistant Element* developed in association with the *Neural Network Agent* gave light to the ability of agents developed under this research to collect information from a simulation environment from which the agents can learn and develop reasoning capabilities during simulation run-time.

The framework developed contributes the creation of a new problem-solving method for CEM issues. This new problem-solving method makes use of agents to improve problem-solving proficiency and relieve modelers of extensive knowledge requirements when developing a simulation model. The agent framework also carries out the resolution process in an automated fashion, which relieves the modeler of the effort required to refine a model from its initial beginnings to a finished product. In addition to creating a new problem-solving method, these two developments in simulation modeling, with respect to the knowledge and effort required to solve problems using simulation, lends to the increased use of simulation modeling for CEM issues. The agent embedded simulation modeling framework removes the negative, laborious, and complexity stigmas from the simulation modeling process. Lastly, the development of the problem solving process provides a contribution towards the future development of tools capable of attaining ideal project management plans.

5.5 Recommendations for Future Development

The first recommendation regarding this research is following through with the development of the agent framework with the goals stated by AbouRizk and Mohammed (2002) in attempt to achieve an "ideal (or perfect) project execution plan". In addition to this, further development can be made with regard to the intelligence available to the agents. This is in reference to developing additional belief networks, neural networks, and algorithms other than those been presented herein. This would provide the agents with capabilities to explore issues outside of what has been previously undertaken. This stated, additional agents could also be constructed that make use of other methods of artificial intelligence reasoning. Lastly, the agent framework could be further investigated to allow for the master/slave agent relationship to be expanded such that master agents could become slave agents supervised by other master agents. This would allow for more detailed embedded agent simulation modeling undertakings where components of a model could fall under different management influences.

References

Abd Majid, M. Z. A., and McCaffer, R. (1998). "Factors of non-excusable delays that influence contractors' performance." *Journal of Management in Engineering*. ASCE, New York, New York, 14(3), 42-48.

AbouRizk, S., and Mohammed, Y. (2002). "Optimal construction project planning." *Proceedings of the 2002 Winter Simulation Conference*. IEEE, New York, New York, 1704 – 1708.

AbouRizk, S. M. (1998). "Value for money audit: project management – cost estimating." *Report to City of Edmonton*. Office of the Auditor General, Edmonton, Alberta.

AbouRizk, S. M. (1993). "Civ.E. 603 – Computer Applications in Construction: Part 3.2 Neural Nets." *Civ.E. 603 Lecture Notes*. Department of Civil and Environmental Engineering, University of Alberta, Edmonton, Alberta.

Ahuja, H. N., Dozzi, S. P., and AbouRizk, S. M. (1994). *Project Management Techniques in Planning and Controlling Construction Projects*. John Wiley & Sons Inc., New York, New York.

Akbulut, M. B., and Sagar, V. K. (2000). "An agent based scheduling system to achieve agility." *Network Intelligence Internet-based Manufacturing Proceedings of SPIE*. SPIE, United States, 4208, 19-30.

Altshuller, G., Ziotin, B., Zusman, A., and Philatov V. (1999). *Tools of Classical TRIZ.* Ideation International Inc. Southfield, MI.

Biswas, S., and Merchawi, S. (2000). "Use of discrete event simulation to validate an agent based scheduling engine." *Proceedings of the 2000 Winter Simulation Conference*. IEEE, New York, New York, 1778-1782.

Bower, J., and Bunn, D. W. (2000). "Model based comparisons of pool and bilateral markets for electricity." *The Energy Journal*. IAEE, Cambridge, Massachusetts, 21(3), 1- 29.

Brennan, R. W., and O, W. (2000). "A simulation test bed to evaluate multi-agent control of manufacturing systems." *Proceedings of the 2000 Winter Simulation Conference*. IEEE, New York, New York, 1747-1756.

Bullock, R. K., McIntyre, G. A., and Hill, R. R. (2000). "Using agent-based modeling to capture airpower strategic effects." *Proceedings of the 2000 Winter Simulation Conference*. IEEE, New York, New York, 1739-1746.

Cares, J. R. (2002). "The use of agent-based models in military concept development." *Proceedings of the 2002 Winter Simulation Conference*. IEEE, New York, New York, 935-939.

Carmicheal, D.G. (1987). *Engineering Queues in Construction and Mining*. Ellis Horwood Limited, Chichester.

Creighton, D., and Nahavandi, S. (2002). "Optimizing discrete event simulation models using a reinforcement learning agent." *Proceedings of the 2002 Winter Simulation Conference*. IEEE, New York, New York, 1945-1950.

Dawid, H., Reimann, M., and Bullnheimer, B. (2001). "To innovate of not to innovate?" *IEEE Transactions on Evolutionary Computation*. IEEE, New York, 5(5), October 2001, 471-481.

Ebben, M. J. R., de Boer, L., and Pop-Sitar, C. E. (2002). "Multi-agent simulation of purchasing activities in organizations." *Proceedings of the 2002 Winter Simulation Conference*. IEEE, New York, New York, 1337-1344.

Farrar, J., Appleton, B., and Shaheen, A. (2001). "*Risk Analysis Summary NEST NC1 Tunnel Project City of Edmonton*." Internal Report through the NSERC/Albert Construction Industry Chair. University of Alberta, Edmonton, Alberta.

Franklin, S., and Graesser, A. (1996). "Is it an agent, or just a program? A taxonomy for autonomous agents." *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*, Springer-Verlag.

Fu, Y., Piplani, R., de Souza, R., and Wu, J. (2000). "Multi-agent enabled modeling and simulation towards collaborative inventory management in supply chains." *Proceedings of the 2000 Winter Simulation Conference*. IEEE, New York, New York, 1763-1771.

Ganeshan, R, Stumpf, A. L., Lui, L.Y., Golish, M., Chin, S., and Hicks, D (1995). "Evaluation of construction time and cost for design alternatives in a collaborative environment." *Proceedings of the 2nd Congress on Computing in Civil Engineering*. ASCE, New York, New York, 2, 1324-1331.

Goldmann, S. (1996). "Procura: a project management model of concurrent planning and design." http://wwwagr.informatik.uni-kl.de/~sigig/Procura/ (accessed on November 21, 2002).

Gu, P., Balasubramanian, S., and Norrie, D. H. (1995). "Bidding-based autonomous planning and scheduling." *Proceedings of SPIE The International Society for Optical Engineering*. SPIE, United States, 2620, 184-194.

Hajjar, D. (1999) . "A Unified Modeling Methodology For Planning Construction Projects." *Doctor of Philosophy Thesis*. University of Alberta, Edmonton, Alberta.

Hajjar, D. and AbouRizk, S. M. (2002). "Unified Modeling Methodology for Construction Simulation." *Journal of Construction Engineering and Management*, ASCE, New York, New York, 128(2), 174-185.

Halpin, D. W., and Riggs L. S. (1992). *Planning and Analysis of Construction Operations.* John Wiley and Sons Inc., New York.

Huang, Y. L., Lui, W., Gou, H., and Wu, C. (2001). "Ontology and multi-agent based decision support for enterprise bidding." *2001 IEEE International Conference on Systems, Man and Cybernetics*, IEEE, New York, New York, 5, 2947- 2951.

International Group for Lean Construction. (1999). http://cic.vtt.fi/lean/index.htm (accessed on May 7, 2003).

Jain, L. C., Chen, Z., and Ichalkaranje, N. (2002). *Intelligent agents and their applications.* Physica-Verlag, New York, New York.

Jones, B., and Riley, M. (1994). "Construction problem solving in a cooperative distributed agent environment." *Computing in Civil Engineering Proceedings of the First Congress Held in Conjunction with A/E/C Systems '94*. ASCE, New York, New York, 1, 79-87.

Khedro, T., Case, M. P., Flemming, U., Genesereth, M. R., Logcher, R., Pederson, C., Snyder, J., Sriram, R. D., and Teicholz, P. M. (1995). "Development of multi-institutional testbed for collaborative facility engineering infrastructure." *Proceedings of the 2nd Congress on Computing in Civil Engineering*. ASCE, New York, New York, 2, 1308-1315.

Khedro, T. (1999). "Collaborative distributed building engineering" *Computer Aided Civil and infrastructure Engineering*. Blackwell Publishers, Malden, MA, 14, 69-79.

Kim, S. K., and Russel, J. S. (2003a). "Framework for an intelligent earthwork system part I system architecture." *Automation in Construction*. Elsevier, New York, 12, 1-13.

Kim, S. K. and Russel, J. S. (2003b). "Framework for an intelligent earthwork system part II Task identification/scheduling and resource allocation methodology." *Automation in Construction*. Elsevier, New York, 12, 15-27.

King, G., Heeringa, B., Westbrook, D., Catalano, J., and Cohen, P. (2002). "Models of defeat." *Proceedings of the 2002 Winter Simulation Conference*. IEEE, New York, New York, 928-934.

Lee, S., Pritchett, A., and Goldsman, D. (2001). "Hybrid agent-based simulation for analyzing the national airspace system." *Proceedings of the 2001 Winter Simulation Conference*. IEEE, New York, New York, 1029-1037.

Mason, C. R., and Moffat, J. (2001). "An agent architecture for implementing command and control in military simulation." *Proceedings of the 2001 Winter Simulation Conference*. IEEE, New York, New York, 721-729.

McCabe, B. Y. (1997). "An automated modeling approach for construction performance improvement using simulation and belief networks." *Doctor of Philosophy Thesis*. University of Alberta, Edmonton, Alberta.

Mizuta, H., and Yamagata, Y. (2001). "Agent based simulation and greenhouse gas emissions trading." *Proceedings of the 2001 Winter Simulation Conference*. IEEE, New York, New York, 535-540.

Mizuta, H., and Steigltz, K. (2000). "Agent-based simulation of dynamic online auctions." *Proceedings of the 2000 Winter Simulation Conference*. IEEE, New York, New York, 1772-1777.

Mohammed, Y. (2002). "A framework for systematic improvement of construction systems." *Doctor of Philosophy Thesis*, University of Alberta, Edmonton, Alberta.

Pitt, J., Anderton, M., and Cunningham, J. (1996). "Normalized interactions between autonomous agents." *The Journal of Collaborative Computing*. Kluwer Academic Publishers, Netherlands, 5, 201-222.

Poole, D.L., Mackworth, A. and Goebel, R. (1998). *Computational Intelligence: A Logical Introduction.* Oxford University Press, New York.

Rasmussen, S. J., and Chandler, P. R. (2002). "MultiUAV: a multiple UAV simulation for investigation of cooperative control." *Proceedings of the 2002 Winter Simulation Conference*. IEEE, New York, New York, 869-877.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks.* Cambridge University Press, United Kingdom.

Ruwanpura J. Y. (2001). "Special Purpose Simulation For Tunnel Construction Operations." *Doctor of Philosophy Thesis,* University of Alberta, Edmonton, Alberta.

Russell, S. and Norvig, P. (1994). *Artificial Intelligence A Modern Approach.* Prentice Hall Inc., New Jersey.

Sanchez, S. M., and Lucas, T. W. (2002). "Exploring the world of agent-based simulations: simple models, complex analyses." *Proceedings of the 2000 Winter Simulation Conference.* IEEE, New York, New York, 116-126.

Schaefer, L. A., Wojcik, L. A., Berry, T. P., and Wanke, C. R. (2002). "Decision support for advanced aviation concepts." *Proceedings of the 2002 Winter Simulation Conference*. IEEE, New York, New York, 1244-1250.

Schaefer, L. A. (2001). "Architecture using jini technology for simulation of an agent-based transportation system." *Proceedings of the 2001 Winter Simulation Conference*. IEEE, New York, New York, 1079-1083.

Scherer, R. J., and Katranuschkov, P. (1995). "Architecture of an object-oriented product model prototype for integrated building design." *Computing In Civil Engineering: Proceedings of the 5th International Conference on Computing in Civil and Building Engineering*. ASCE, New York, New York, 393-400.

Shi-ping, L., Yun-qing, R., Jie, Z., and Pei-gen, L. (2002). "Model of production planning and scheduling in a shop floor based on collaboration and competition." *Wuhan University Journal of Natural Sciences*. Wuhan, Daxue, China, 7(1) 77-81.

Szymanski, B., and Chen, G. (2000). "Linking spatially explicit parallel continuous and discrete models." *Proceedings of the 2000 Winter Simulation Conference*. IEEE, New York, New York, 1705-1712.

Tassier, T., and Menczer, F. (2001). "Emerging small-world referral networks in evolutionary labour markets." *IEEE Transactions on Evolutionary Computation*. IEEE, New York, 5(5), October 2001, 482-492.

Teredesai, T., and Ramesh, V. C. (1998). "A multi-agent mixed initiative system for real-time scheduling." *1998 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, New York, New York, 1, 439-444.

Tommelein, I. D. (1998). "Pull driven scheduling for pipe-spool installation: simulation of lean construction technique." *Journal of Construction Engineering and Management*. ASCE, New York, New York, 124(4), 279-288.

Tserng, H. P., and Lin, W. Y. (2003). "Developing an electronic acquisition model for project scheduling using XML-based information standard." *Automation in Construction*. Elsevier, New York, 12, 67-95.

Turban, E., and Aronson, J. E. (2001). *Decision support systems and intelligent systems*. Prentice Hall, Upper Saddle River, New Jersey.

Van Tol, A., AbouRizk, S., Er, K. C., and Fernando, S. (2002). "Boring deeper into tunneling simulation." *NARMS-TAC 2002 Mining and Tunnelling Innovation and Opportunity*. University of Toronto Press, Toronto, 2, 1697-1703.

Van Tol, A. A. (2000). "Descriptive simulation modeling with resource allocation using belief networks." *Masters of Science Thesis*, University of Alberta, Edmonton, Alberta.

Van Tol, A. A. and AbouRizk, S. M. (TBA) "Agent Embedded Simulation Modeling" *Submitted to Journal of Construction Engineering and Management*. ASCE, New York, New York.

Wales, R. J. (1994). "Incorporating weather effects in project simulation." *Masters of Science Thesis*, University of Alberta, Edmonton, Alberta.

Weiss, G. (1999). *Multiagent systems: a modern approach to distributed artificial intelligence*. The MIT Press, Cambridge, Massachusetts.

Whelan, M., Loftus, J., Perme, D., and Baldwin, R. (2002). "Reducing training costs through integration of simulations, C4I systems, and expert systems." *Proceedings of the 2002 Winter Simulation Conference*. IEEE, New York, New York, 940-947.

Wilson, L. F., Burroughs, D., Sucharitaves, J., and Kumar, A. (2000). "An agent-based framework for linking distributed simulations." *Proceedings of the 2000 Winter Simulation Conference*. IEEE, New York, New York, 1713-1721.

Womack, J. P., and Jones, D.T. (1996). *Lean Thinking*. Simon & Schuster, New York, New York.

Xiong, G., Hashiyama, T., and Okuma, S. (2002). "An evolutionary computation for supplier bidding strategy in electricity auction market." *Transactions of The Institute of Electrical Engineers of Japan*. Inst. of Electrical Engineers of Japan, Japan, 122, 1830-1836.

APPENDIX 1 – Borehole Database

A1.1 Borehole Database Instructions

The borehole database program allows one to compile and access borehole log data. The program itself is a Microsoft Visual Basic program that makes use of a Microsoft Access Database for data storage. Operation of the database is carried out through a single display screen as shown in Figure A1-1. The display screen, or form, is divided into two sections. The top section consists of text boxes that are used to capture general information about the borehole log such as the projects name, date the hole was bored, and so on. The bottom portion of the form consists of the "Borehole Details" grid. This grid contains the specific properties for each individual soil segment to be found in the borehole log (e.g. soil location, composition, water content, and so on). The form presented to the user allows for only four possible actions to be carried out.

A1.1.1 Scrolling The Database

To scroll the database the user simply uses the "Borehole log" scroll device located to the upper right of the Borehole Details grid. Clicking on the right arrow button of the scroll device advances the database to the next borehole in the database. Clicking on the left arrow button of the scroll device takes the program back to the previous borehole in the database.

A1.1.2 Editing Data

To edit the data previously entered into the database program the user simply scrolls to the borehole log they wish to edit and begin editing the information as it is presented on the form. Adding another row of information to the "Borehole Details" grid is accomplished by clicking on the first empty row at the bottom of the grid, which is located below the last detailed row in the grid, and entering data. Rows can be removed in the grid by clicking on the grey box on the very left side of the row, such that the entire row is highlighted, and pressing delete.

A1.1.3 Adding Data

The user can add borehole logs to the database by clicking the "Add Borehole" button. This action will add another record to the database and clear form so that the user can begin entering data. The general data entered in the text boxes in the top of the form are as follows:

Project – The projects name

Date Bored – The date the borehole was bored

Bore Depth – The depth of the borehole

Hole Elevation – The elevation at the top of the borehole

Depth To Water Table – The depth from the top of the borehole to the water table

North - South Location Description – A common geographic description to help aid in defining the location of the borehole. (e.g. a street reference)

Northing – A Numerical value, as used in surveying, that can be used to reference the location of the current borehole to other boreholes or geographic locations

East – West Location Description - A common geographic description to help aid in defining the location of the borehole. (e.g. a street reference)

Easting - A Numerical value, as used in surveying, that can be used to reference the location of the current borehole to other boreholes or geographic locations

The information in the borehole details grid relates to the specific information for each soil layer found in the borehole. The information entered in this grid is as follows:

Start Depth – The top elevation for the soil layer

End Depth – The bottom elevation for the soil layer

Soil Description – A brief description of the soil

Soil Type – A drop down menu where the user specifies the soil as one of 11 possible types

Comments – A location for any comments about the soil to be given

Color – A geotechnical description of the soils color

Granularity - A geotechnical description of the soils granularity

Plasticity - A geotechnical description of the presence of sand pockets within the soil

Pockets - A description of the soils color

CS - A geotechnical description of the soils compressive strength

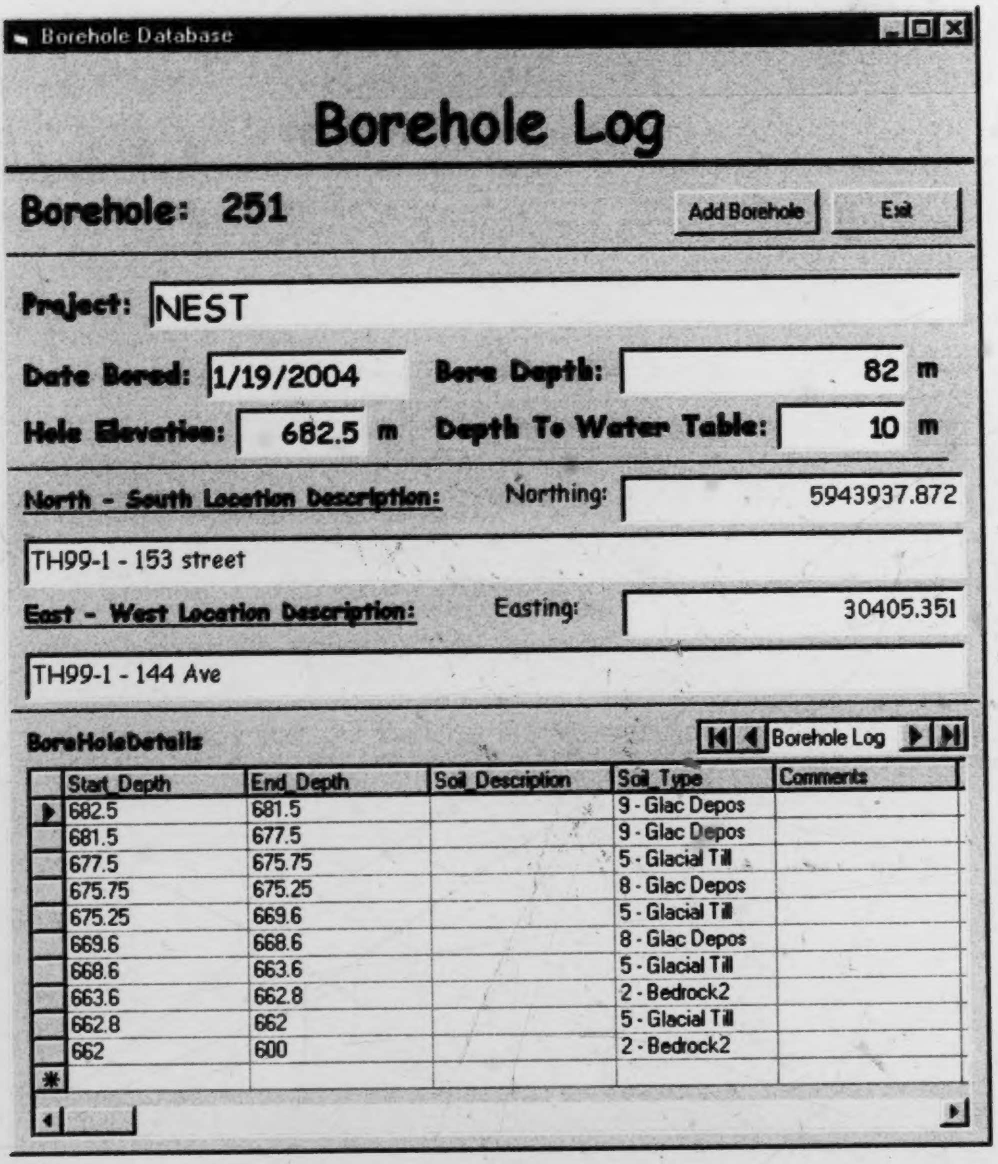
SPT - A geotechnical description of the soils Standard Penetration Test

WC - A geotechnical description of the soils water content

The number of rows to be used in the grid is unlimited. The user simply clicks on the first row to begin entering data. Additional rows are entered by clicking on the subsequent row until all the data has been entered. Rows can be removed in the grid by clicking on the grey box on the very left side of the row, such that the entire row is highlighted, and pressing delete.

A1.1.4 Exiting the Program

The user can exit the program using the “Exit” button.


Borehole Database
Borehole Log
Add Borehole
Exit
BoreHoleDetails
Borehole Log

Borehole: 251

Project: NEST

Date Bored: 1/19/2004

Bore Depth: 82 m

Hole Elevation: 682.5 m

Depth To Water Table: 10 m

North - South Location Description: TH99-1 - 153 street

Northing: 5943937.872

East - West Location Description: TH99-1 - 144 Ave

Easting: 30405.351

Start_Depth	End_Depth	Soil_Description	Soil_Type	Comments
682.5	681.5		9 - Glac Depos	
681.5	677.5		9 - Glac Depos	
677.5	675.75		5 - Glacial Till	
675.75	675.25		8 - Glac Depos	
675.25	669.6		5 - Glacial Till	
669.6	668.6		8 - Glac Depos	
668.6	663.6		5 - Glacial Till	
663.6	662.8		2 - Bedrock2	
662.8	662		5 - Glacial Till	
662	600		2 - Bedrock2	

Figure A1-1. Borehole Database Program Form

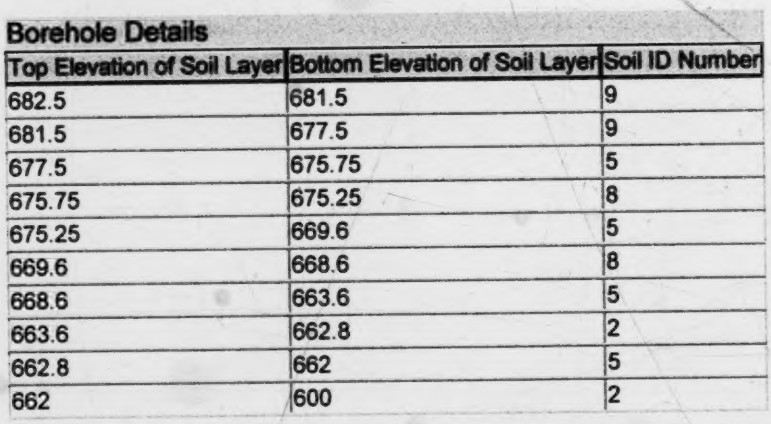
A1.2 NEST Borehole Data

A1.2.1 Bore Hole: TH99-1

Northing: 5943937.872

Easting: 30405.351

Table A1-1. Data For Borehole TH99-1



Borehole Details

Top Elevation of Soil Layer	Bottom Elevation of Soil Layer	Soil ID Number
682.5	681.5	9
681.5	677.5	9
677.5	675.75	5
675.75	675.25	8
675.25	669.6	5
669.6	668.6	8
668.6	663.6	5
663.6	662.8	2
662.8	662	5
662	600	2

A1.2.2 Bore Hole: TH00-2

Northing: 5943902.872

Easting: 30495.351

Table A1-2. Data For Borehole TH00-2

Borehole Details		
Top Elevation of Soil Layer	Bottom Elevation of Soil Layer	Soil ID Number
681.75	681.6	9
681.6	678	9
678	672.4	5
672.4	671.9	8
671.9	667.5	5
667.5	665.5	2
665.5	661.4	5
661.4	600	2

A1.2.3 Bore Hole: TH99-2

Northing: 5943935.372

Easting: 30807.851

Table A1-3. Data For Borehole TH99-2

Borehole Details		
Top Elevation of Soil Layer	Bottom Elevation of Soil Layer	Soil ID Number
679.6	678.7	5
678.7	674.7	5
674.7	666.8	5
666.8	665.2	9
665.2	663.4	5
663.4	662.4	5
662.4	660.4	5
660.4	600	8

A1.2.4 Bore Hole: TH00-4

Northing: 5943905.372

Easting: 30930.351

Table A1-4. Data For Borehole TH00-4

Borehole Details		
Top Elevation of Soil Layer	Bottom Elevation of Soil Layer	Soil ID Number
679.5	679.25	5
679.25	675.7	5
675.7	663.6	5
663.6	661.5	8
661.5	600.6	2
600.6	600	5

A1.2.5 Bore Hole: TH99-3

Northing: 5943932.872

Easting: 31087.851

Table A1-5. Data For Borehole TH99-3

Borehole Details		
Top Elevation of Soil Layer	Bottom Elevation of Soil Layer	Soil ID Number
680	679.4	5
679.4	675.25	5
675.25	661	5
661	600	2

A1.2.6 Bore Hole: TH00-3

Northing: 5943862.872

Easting: 31222.851

Table A1-6. Data For Borehole TH00-3

Borehole Details		
Top Elevation of Soil Layer	Bottom Elevation of Soil Layer	Soil ID Number
678.3	678.1	9
678.1	674.6	9
674.6	660.4	5
660.4	659.8	8
659.8	659.4	5
659.4	658.8	2
658.8	600	2

A1.2.7 Bore Hole: TH00-1

Northing: 5943642.872

Easting: 31790.351

Table A1-7. Data For Borehole TH00-1

Borehole Details		
Top Elevation of Soil Layer	Bottom Elevation of Soil Layer	Soil ID Number
678.8	678	9
678	672.5	9
672.5	659.6	5
659.6	658.5	2
658.5	600	5

A1.2.8 Bore Hole: TH1-5

Northing: 5943707.872

Easting: 31905.351

Table A1-8. Data For Borehole TH1-5

Borehole Details

Top Elevation of Soil Layer	Bottom Elevation of Soil Layer	Soil ID Number
670.25	666.25	5
666.25	600	5

A1.2.9 Bore Hole: TH99-4

Northing: 5943765.372

Easting: 31452.851

Table A1-9. Data For Borehole TH99-4

Borehole Details

Top Elevation of Soil Layer	Bottom Elevation of Soil Layer	Soil ID Number
678.3	678.2	5
678.2	674.6	5
674.6	660.4	5
660.4	659.8	8
659.8	659.4	2
659.4	658.8	8
658.8	600	2

A1.3 Borehole Database Overview

The Borehole Database is Visual Basic program that makes use of an Access database program to electronically store and retrieve geotechnical borehole log records. The Visual Basic program consists of one form as shown in Figure A1-2. The form was constructed using the common Visual Basic components as well as the Microsoft Data bound List Controls, APEX True DBGrid Pro 6.0, and APEX True DBList Pro 6.0 add on components. The APEX True DBGrid Pro 6.0 add on components were used to provide for the grid feature located at the bottom of the form. The APEX True DBList Pro 6.0 add on components was used to provide for a drop down list box for the soil type selection within the grid. This drop down feature was selected because the soil types were defined in the database file by 11 specific categories.

The database is a relational database constructed using 3 Tables as shown in Figure A1-3. The "Boreholes" table contained the general information for each borehole log in the database. It was constructed with 11 fields. The information regarding these fields is as follows:

BoreHole_ID – This is the primary key for the table. The data type is defined as an auto number. This field is used to uniquely identify the individual borehole records in the table.

Borehole_No –The data type is defined as a number. This field is used to record the borehole number in the database.

Project - The data type is defined as a text field. This field is used to record the project name associated with the borehole log.

EWLocation - The data type is defined as a text field. This field is used to record an east-west street location or some other description that would be commonly known.

NSLocation - The data type is defined as a text field. This field is used to record a north-south street location or some other description that would be commonly known.

Easting - The data type is defined as a number. This field is used to record the easting coordinate for the borehole location.

Northing - The data type is defined as a number. This field is used to record the northing coordinate for the borehole location.

Date - The data type is defined as a date/time field. This field is used to record the date when the borehole was bored.

Elevation - The data type is defined as a number. This field is used to record the top elevation of the borehole.

Bore_Depth - The data type is defined as a number. This field is used to record the depth to which the borehole was bored.

Water_Table_Depth - The data type is defined as a number. This field is used to record the depth where the water table was discovered.

The "Soil" table contained the specific categories of soil types to be used in the database. The table contained 3 fields. Their information is as follows:

Soil_ID – This is the primary key for the table. The data type is defined as an auto number. This field is used to uniquely identify the individual soil records in the table.

Soil_No –The data type is defined as a number. This field is used to record the assigned number for the soil type in the database.

Description - The data type is defined as a text field. This field is used to record a general description for the soil that could be commonly understood.

The "Borehole_Details" table contains the specific information for the soil layers found in the borehole logs. Therefore this table has fields which relate it back to the "Boreholes" and "Soil" soil tables.

Detail_ID – This is the primary key for the table. The data type is defined as an auto number. This field is used to uniquely identify the individual borehole detail records in the table.

Borehole_ID –The data type is defined as a number. This is a reference to the primary key in the Borehole table.

Start_Depth –The data type is defined as a number. This field records the start depth for a soil layer in the borehole log.

End_Depth –The data type is defined as a number. This field records the start depth for a soil layer in the borehole log.

Soil_Description - The data type is defined as a text field. This field records a description given for the soil layer.

Soil_ID –The data type is defined as a number. This is a reference to the primary key in the Soil table.

Comments - The data type is defined as a text field. This filed allows for general comment to be entered for the soil layer.

Color - The data type is defined as a text field. This field provide for a description of the soils color.

Granularity - The data type is defined as a text field. This field provide for a description of the soils granularity.

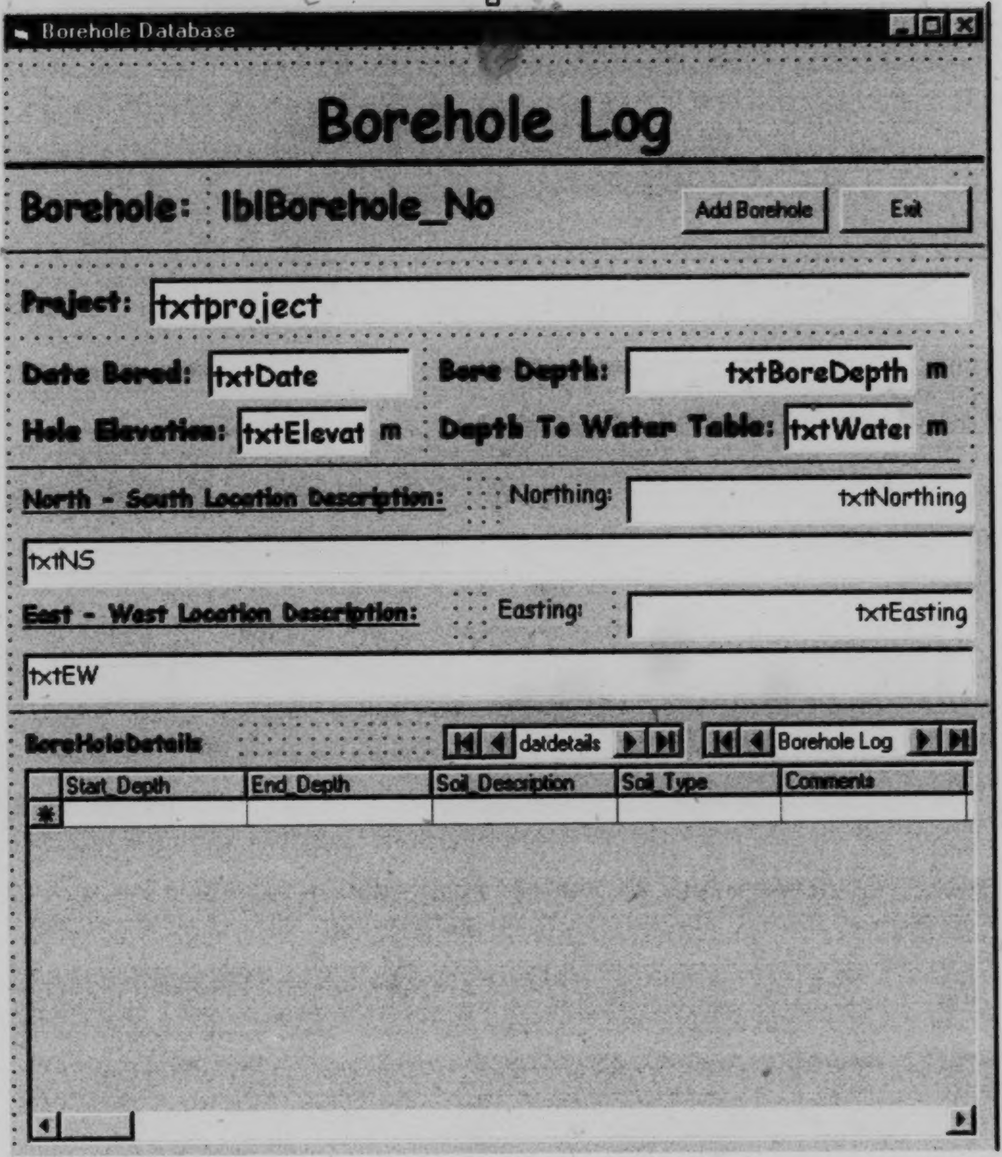
Plasticity - The data type is defined as a text field. This field provide for a description of the soils plasticity.

Pockets - The data type is defined as a text field. This field provide for a description of the presence of sand pockets in the soil.

CS –The data type is defined as a number. This field records the soils compressive strength.

SPT –The data type is defined as a number. This field records the soils Standard Penetration Test results.

WC –The data type is defined as a number. This field records the soils water content.


Borehole Database
Add Borehole
Exit
datdetails
Borehole Log

Borehole Log

Borehole: lblBorehole_No

Project: txtproject

Date Bored: txtDate

Bore Depth: txtBoreDepth m

Hole Elevation: txtElevat m

Depth To Water Table: txtWater m

North - South Location Description: txtNS

Northing: txtNorthing

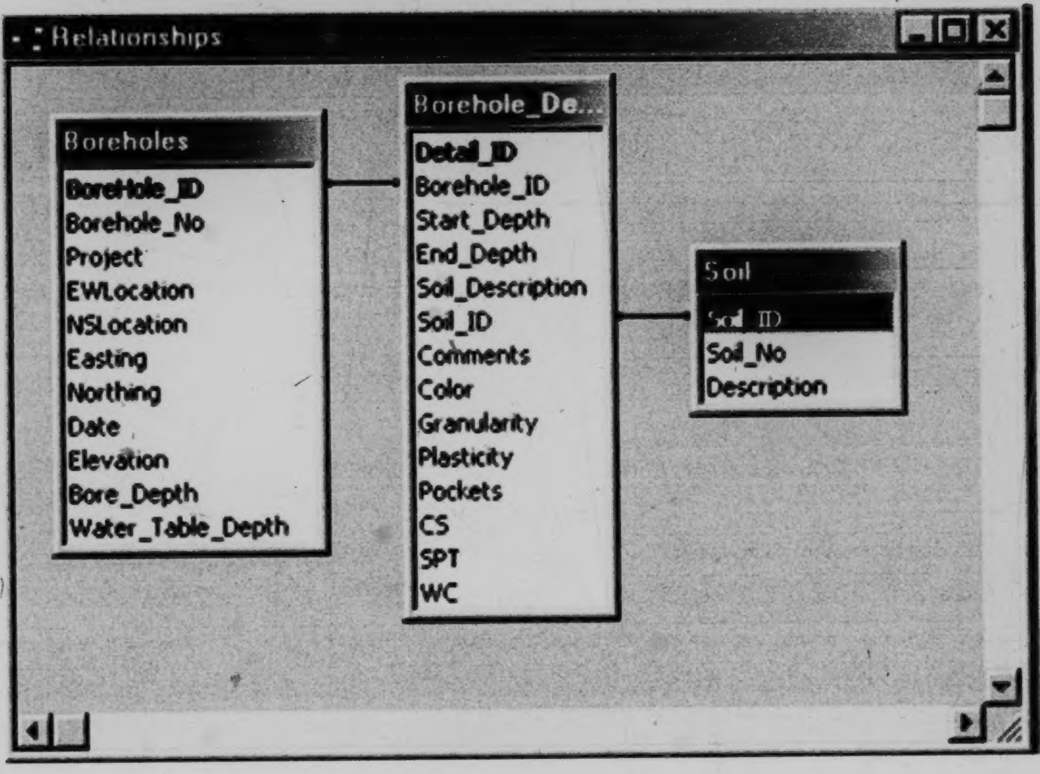
East - West Location Description: txtEW

Easting: txtEasting

BoreHoleDetails

Start_Depth	End_Depth	Soil_Description	Soil_Type	Comments

Figure A1-2. Borehole Database Program Form Design View



Relationships

Boreholes
BoreHole_ID
Borehole_No
Project
EWLocation
NSLocation
Easting
Northing
Date
Elevation
Bore_Depth
Water_Table_Depth

Borehole_De...
Detail_ID
Borehole_ID
Start_Depth
End_Depth
Soil_Description
Soil_ID
Comments
Color
Granularity
Plasticity
Pockets
CS
SPT
WC

Soil
Soil_ID
Soil_No
Description

Figure A1-3. Borehole Database Relational Diagram

A1.4 Database Program Code

The following is the Visual Basic Code used to construct the Borehole database program.

```
Public BoreholeID As Integer
Private Sub cmdAddBorehole_Click()

   'add a new borehole to the database
   datBoreHole.Recordset.AddNew
   BoreholeID = datBoreHole.Recordset!Borehole_ID
   datBoreHole.Recordset!BoreHole_No = BoreholeID

   'default data
   datBoreHole.Recordset!Project = "Project Name"
   datBoreHole.Recordset!EWLocation = "Geographic Location"
   datBoreHole.Recordset!NSLocation = "Geographic Location"
   datBoreHole.Recordset!Easting = 0
   datBoreHole.Recordset!Northing = 0
   datBoreHole.Recordset!Date = Format(Now(), "dd-mmm-yy")
   datBoreHole.Recordset!Elevation = 0
   datBoreHole.Recordset!Bore_Depth = 0
   datBoreHole.Recordset!Water_Table_Depth = 0
   datBoreHole.Recordset.Update
   datDetails.Refresh
   datBoreHole.Recordset.MoveLast
   datDetails.Refresh

End Sub
```

```
'end the program
Private Sub cmdEnd_Click()
   End
End Sub
```

```
'call sub datarefresher that updates details grid
Private Sub DatBoreHole_Reposition()
   DataRefresher
End Sub
```

```
'here's the sql statement for the grid
Public Sub DataRefresher()
   Dim SQLStat As String
   SQLStat = "SELECT * FROM Borehole_Details WHERE
(((Borehole_details.BoreHole_ID)=" & datBoreHole.Recordset!Borehole_ID & "));"
    datDetails.RecordSource = SQLStat
   datDetails.Refresh
End Sub
```

```
Private Sub Form_Load()
    'this is what loads the soil type info from the database into the drop down box in the grid
    Dim vi As ValueItem
    Dim rs As Recordset
    Dim db As Database

    Set db = OpenDatabase("c:/borehole/borehole.mdb")
    Set rs = db.OpenRecordset("SELECT Soil.* FROM Soil;")
    rs.MoveFirst
    Do Until rs.EOF
        Set vi = New ValueItem
        vi.Value = rs!soil_id
        vi.DisplayValue = rs!soil_no & " - " & rs!Description
        grdDetails.Columns("Soil_Type").ValueItems.Add vi
        rs.MoveNext
    Loop
    rs.Close
    db.Close
End Sub
```

```
Private Sub grdDetails_OnAddNew()
    'this sets the borehole id number for the details records in the database that are added via a
new row in the grid
```

```
    grdDetails.Columns("Borehole_ID").Value = datBoreHole.Recordset!Borehole_ID

End Sub
```

APPENDIX 2 – Neural Network Database

A2.1 Neural Network Database Instructions

The neural network database program allows one to compile and access the neural network data that are used by the neural network agent to design, train and recall neural networks. The program was constructed as a Microsoft Visual Basic program that makes use of a Microsoft Access Database for data storage. Data manipulation within the database is carried out through two program forms as shown in Figure A2-1 and A2-2.

A2.1.1 The Main Form

The first form presented to the user when running the program is the main form (Figure A2-1). This form lists all the neural network data files stored within the database in the display grid. The form has three possible actions that can be carried out by the user.

A2.1.1.1 Creating A New Neural Network File

A new network file can be added to the database by using the "Create New Net" button on the main form. After clicking the "Crate New Net" button the user will be presented with a series of input boxes posing questions that will create the new neural network file in the database. The first input box asks the user to provide a name for the network file so that it can be identified to the user. The names are then listed in the display grid on the main form. Thus from figure A2-1 it can be seen that one such network file has been named "Weather". The next input box asks the user for the number on input variables found in each data vector for the neural network. In the case of the "Weather" network the data vectors consist of three input variables (Average Daily Temperature, Precipitation, and 7-Day Cumulative Precipitation) and one output variable (Productivity Factor). The user would respond to the input box with a response of "3". The following input box would prompt the user for the number of output variables to which the response would be "1". The following input boxes would ask the user to provide descriptive labels for the input and output vector variables. The user would provide responses similar to "Average Daily Temperature", "Precipitation", "7-Day Cumulative Precipitation" and "Productivity Factor" so that a descriptive reference can be given to the variables. At the conclusion of the definition of the variable descriptions the user would then be returned to the main form and the new network file would be displayed in the display grid.

A2.1.1.2 Edit Data Within An Existing Neural Network File

Data can be edited or added to a neural network file by clicking on the file name in the display grid on the main form and then clicking on the "Edit Data" button. This action will direct the user to the Data Editing Form as show in Figure A2-2.

A2.1.1.3 Exiting The Program

The user can exit the program using the "Exit" button.

A2.1.2 The Data Editing Form

The data editing form (Figure A2-2) is presented to the user after the "Edit Data" button in the main form has been used. The form is the location in the program where the user can edit data previously entered into the database or add new data. The form has three possible actions that can be carried out by the user.

A2.1.2.1 Edit Existing Data

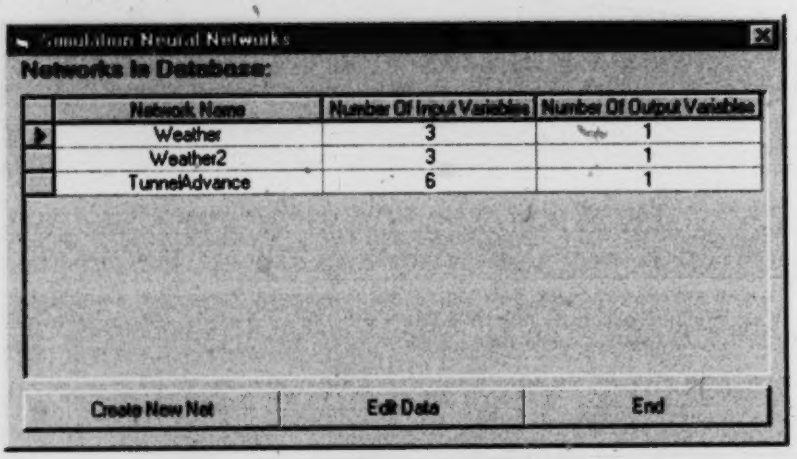
Data previously entered in the database can be readily edited by clicking on the data in question where it is found in the display grid. The data can then be corrected by simply typing the new information over the old.

A2.1.2.2 Add New Data

New data can be added to the network file by clicking on the "Add Vector" button. After clicking on the "Add Vector" button the user will be presented with a series of input boxes prompting the user for the input and output values in the order as they are listed in the display grid. At the completion of the data entry the user will be returned to the data entry form and the new data vector will appear in the display grid.

A2.1.2.3 Return To The Main Form

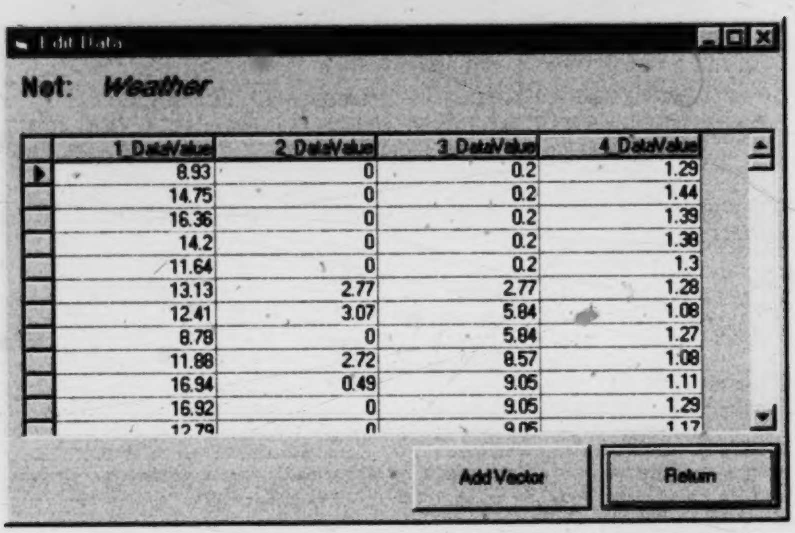
The user returns to the main form by using the "Return" button.


Simulation Neural Networks
Create New Net
Edit Data
End

Networks In Database:

Network Name	Number Of Input Variables	Number Of Output Variables
Weather	3	1
Weather2	3	1
TunnelAdvance	6	1

Figure A2-1. Neural Network Database Program Main Form


Edit Data
Add Vector
Return

Net: *Weather*

1_DataValue	2_DataValue	3_DataValue	4_DataValue
8.93	0	0.2	1.29
14.75	0	0.2	1.44
16.36	0	0.2	1.39
14.2	0	0.2	1.38
11.64	0	0.2	1.3
13.13	2.77	2.77	1.28
12.41	3.07	5.84	1.08
8.78	0	5.84	1.27
11.88	2.72	8.57	1.08
16.94	0.49	9.05	1.11
16.92	0	9.05	1.29
12.79	0	9.05	1.17

Figure A2-2. Neural Network Database Program Data Editing Form

Neural Network Weather Data

The information presented in this section are the data vectors used to define the weather neural network presented in this thesis. This information has been taken form Wales (1994)

Table A2-1. Data For Weather Neural Network

Average Daily Temperature	Precipitation (mm)	Precipitation Sum 7 Days Previous (mm)	Productivity Factor
8.93	0.00	0.20	1.29
14.75	0.00	0.20	1.44
16.36	0.00	0.20	1.39
14.20	0.00	0.20	1.38
11.64	0.00	0.20	1.30
13.13	2.77	2.77	1.28
12.41	3.07	5.84	1.08
8.78	0.00	5.84	1.27
11.80	2.72	8.57	1.08
16.94	0.49	9.05	1.11
16.92	0.00	9.05	1.29
12.79	0.00	9.05	1.17
16.45	0.00	6.28	1.16
14.53	0.00	3.21	1.23
16.66	0.00	3.21	1.20
16.35	0.00	0.49	1.37
21.57	0.00	0.00	1.33
23.05	0.00	0.00	1.30
21.06	0.00	0.00	1.27
18.68	0.00	0.00	1.38
15.17	13.94	13.94	0.00
11.54	0.00	13.94	1.11
11.71	0.00	13.94	1.15
16.08	9.86	23.80	0.78
15.69	0.56	24.36	1.12
14.46	0.63	24.99	1.09
12.94	2.70	27.70	0.90
14.66	0.00	13.76	1.30
16.49	0.86	14.62	1.06
16.43	0.00	14.62	1.06
12.15	0.25	5.01	1.22
13.92	12.03	16.48	0.00
13.01	12.13	27.98	0.00
12.26	0.73	26.00	0.96

Average Daily Temperature	Precipitation (mm)	Precipitation Sum 7 Days Previous (mm)	Productivity Factor
8.44	1.27	27.27	0.88
8.97	10.26	36.66	0.61
11.23	0.00	36.66	0.85
13.98	0.00	36.42	0.97
14.07	2.62	27.01	0.97
12.72	1.12	16.00	1.20
12.86	9.81	25.08	0.77
17.98	0.00	23.81	1.09
15.13	0.32	13.87	1.11
19.84	0.00	13.87	1.20
19.30	1.59	15.46	1.17
18.14	0.00	12.85	1.20
16.70	9.12	20.84	0.79
20.68	0.00	11.04	1.21
18.15	0.00	11.04	1.27
19.78	0.00	10.71	1.20
19.57	0.00	10.71	1.14
16.55	2.56	11.68	1.08
14.56	10.55	22.24	0.76
22.09	0.00	13.12	1.25
25.26	3.51	16.63	0.94
20.71	10.46	27.08	0.63
25.68	15.76	42.84	0.00
23.25	0.15	42.99	0.97
17.46	13.95	54.37	0.00
19.67	10.00	53.82	0.00
13.24	0.89	54.71	0.00
17.70	5.24	56.44	0.00
21.74	3.08	49.07	0.00
23.50	0.00	33.31	1.04
24.27	0.00	33.17	0.97
22.10	0.00	19.22	1.19
23.42	0.00	9.22	1.21
20.57	10.98	19.30	0.63
27.10	0.00	14.06	1.16
26.37	0.00	10.98	1.17
19.57	13.21	24.19	0.00
18.35	0.21	24.40	1.07
14.76	2.01	26.41	1.09
19.70	21.64	48.05	0.00
23.87	0.00	37.07	0.88
21.74	0.22	37.30	0.90
20.50	16.22	53.52	0.00
22.50	0.00	40.31	0.81
16.59	1.40	41.50	0.87
17.35	1.05	40.54	0.91
16.14	0.11	19.01	1.07

Average Daily Temperature	Precipitation (mm)	Precipitation Sum 7 Days Previous (mm)	Productivity Factor
17.43	5.27	24.28	0.91
16.41	0.47	24.54	1.10
13.98	2.67	10.98	1.22
20.24	3.90	14.88	0.98
20.03	1.14	14.62	1.19
18.54	0.47	14.04	1.24
19.48	0.00	13.92	1.18
19.38	0.00	8.65	1.30
20.32	0.00	8.18	1.13
22.90	0.00	5.51	1.25
21.47	0.68	2.29	1.31
22.05	0.94	2.09	1.25
19.45	4.54	6.16	1.12
15.62	12.28	18.44	0.00
24.44	0.00	18.44	1.02
25.28	0.00	18.44	1.06
26.17	0.00	18.44	1.22
19.46	1.77	19.53	0.99
25.77	0.00	18.59	1.05
21.90	0.00	14.05	1.17
20.42	0.00	1.77	1.33
26.87	0.00	1.77	1.22
23.79	0.14	1.91	1.21
20.94	12.84	14.75	0.00
19.62	0.00	12.98	1.27
15.48	0.32	13.30	1.19
19.48	0.00	13.30	1.12
15.88	0.00	13.30	1.15
18.87	0.00	13.30	1.22
14.78	2.06	15.22	1.15
12.43	2.16	4.54	1.28
15.83	0.00	4.54	1.30
17.49	0.00	4.22	1.24
15.79	0.00	4.22	1.17
14.82	0.00	4.22	1.38
12.25	10.88	15.10	0.73
10.64	0.00	13.04	1.14
16.63	0.00	10.88	1.22
23.41	0.00	10.88	1.23
14.99	2.90	13.78	1.20
18.02	1.51	15.29	1.08
12.24	7.32	22.61	0.84
11.56	0.02	11.75	1.17
11.18	5.13	16.88	1.00
10.95	2.83	19.70	1.03
14.99	0.00	19.70	1.24
16.22	0.00	16.80	1.09

Average Daily Temperature	Precipitation (mm)	Precipitation Sum 7 Days Previous (mm)	Productivity Factor
14.56	0.00	15.30	1.15
12.27	0.00	7.98	1.32
8.89	20.28	28.23	0.00
9.86	8.89	31.99	0.67
17.18	0.00	29.17	1.05
14.48	6.56	35.73	0.82
17.05	1.08	36.81	0.89
12.68	3.68	40.49	0.86
14.11	4.18	44.67	0.68
13.15	0.00	24.40	1.02
14.80	0.00	15.50	1.17
13.99	0.00	15.50	1.17
11.95	0.00	8.94	1.30
9.27	0.00	7.86	1.18
9.41	0.00	4.18	1.32
16.03	0.00	0.00	1.27
15.88	0.00	0.00	1.23
13.21	0.00	0.00	1.34
13.89	0.00	0.00	1.31
14.97	0.00	0.00	1.29
16.19	0.00	0.00	1.22
7.12	0.00	0.00	1.29
3.38	0.00	0.00	1.24
4.22	1.05	1.05	1.25
8.00	0.00	1.05	1.22
9.09	0.00	1.05	1.34
8.98	0.00	1.05	1.19
11.17	0.00	1.05	1.33
8.56	3.74	4.79	1.12
13.94	0.00	4.79	1.38
16.21	0.00	3.74	1.36
17.27	0.00	3.74	1.26
11.74	0.00	3.74	1.20
15.04	0.00	3.74	1.33
22.15	0.00	3.74	1.32
16.38	0.00	0.00	1.25
13.91	0.00	0.00	1.42
9.83	0.00	0.00	1.33
2.65	0.00	0.00	1.21
6.82	0.00	0.00	1.18
6.79	0.00	0.00	1.25
2.35	0.00	0.00	1.12
2.67	0.00	0.00	1.21
4.46	0.00	0.00	1.29
-2.65	0.33	0.33	1.18
3.76	0.00	0.33	1.27
0.41	0.74	1.07	1.12

Average Daily Temperature	Precipitation (mm)	Precipitation Sum 7 Days Previous (mm)	Productivity Factor
2.88	0.00	1.07	1.19
-4.26	0.00	1.07	1.09
-2.21	0.25	1.32	1.20
-1.01	7.94	9.26	0.70
0.00	0.00	8.93	1.08
7.78	0.00	8.93	1.24
6.03	0.00	8.19	1.21
2.25	0.00	8.19	1.17
-6.53	0.77	8.96	0.90
0.76	0.00	8.71	1.18
1.85	0.00	0.77	1.19
1.52	0.00	0.77	1.25
-3.01	0.00	0.77	1.21
-4.32	0.00	0.77	1.11
-12.76	4.25	5.02	0.78
-3.42	0.00	4.25	1.05
-3.57	0.00	4.25	1.06
-1.31	0.00	4.25	1.12
-10.27	2.08	6.33	0.81
-8.18	0.00	6.33	0.95
1.66	0.00	6.33	1.17
2.19	0.43	2.51	1.06
4.69	1.00	3.50	1.06
8.45	0.00	3.50	1.21
8.38	0.00	3.50	1.21
0.43	0.18	1.61	1.12
-6.28	0.00	1.61	1.14
-4.64	0.00	1.61	1.18
-7.01	0.00	1.18	0.98
-11.20	1.50	1.67	0.91
-13.78	10.47	12.14	0.51
-11.07	0.00	12.14	0.87
-10.91	0.00	11.96	0.97
-2.79	0.00	11.96	1.07
-6.85	0.00	11.96	0.87
-9.98	0.00	11.96	0.92
-9.75	0.00	10.47	1.00
-3.85	0.00	0.00	1.16
4.76	0.00	0.00	1.12

A2.2 Neural Network Database Overview

The Neural Network Database is a Microsoft Visual Basic program that makes use of a Microsoft Access database program to electronically store and retrieve the historical data required for developing and training a neural network. The database program consists of two forms that allow the user to create neural network files with in a database file and store historical vector data within those files (Figures A2-3 and A2-4). The forms were constructed using the common Visual Basic components supplemented with the APEX True DBGrid Pro 6.0 add on component.

The database is a relational database constructed with 3 Tables as presented in Figure A2-5. The "Net" table contained the general information that defined each neural network folder within the database. It was constructed with 4 fields. The information regarding these fields is as follows:

NetID – This is the primary key for the table. The data type is defined as an auto number.

NetName - The data type is defined as a text field. This field is used to record the name associated with the neural network.

InputNum - The data type is defined as a number. This field is used to record the number of input variables found in each data vector.

OutputNum – The data type is defined as a number. This field is used to record the number of output variables found in each data vector.

The "NetDataInfo" table contained the information that described each data value found within the data vectors for the neural network. It was constructed with 4 fields. The information regarding these fields is as follows:

NetDataInfoID - This is the primary key for the table. The data type is defined as an auto number.

NetID - The data type is defined as a number. This is a reference to the primary key in the Net table.

DataType - The data type is defined as a text field. This field is used to record the type of data being represented in the vector (e.g. input1, input2, output1 ...).

TypeName - The data type is defined as a text field. This field is used to record a user defined label given to describe the number value within the vector (e.g. temperature).

The "NetData" table contained the information for the stored data vectors within the neural network database file. It was constructed with 4 fields. The information regarding these fields is as follows:

NetDataID - This is the primary key for the table. The data type is defined as an auto number.

NetDataInfoID - The data type is defined as a number. This is a reference to the primary key in the NetDataInfo table.

RecordNo - The data type is defined as a number. This field is used to record the vector number that is associated with all the input and output variables for a specific vector.

DataValue – The data type is defined as a number. This field is used to record the value for a specific number found within the vector.

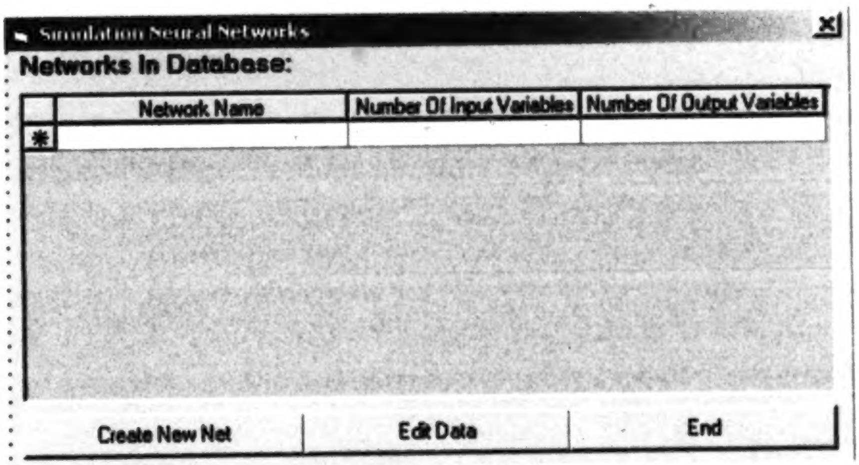

Simulation Neural Networks
Networks In Database:
Network Name
Number Of Input Variables
Number Of Output Variables
Create New Net
Edit Data
End

Figure A2-3. Neural Network Database Program Main Form Design View

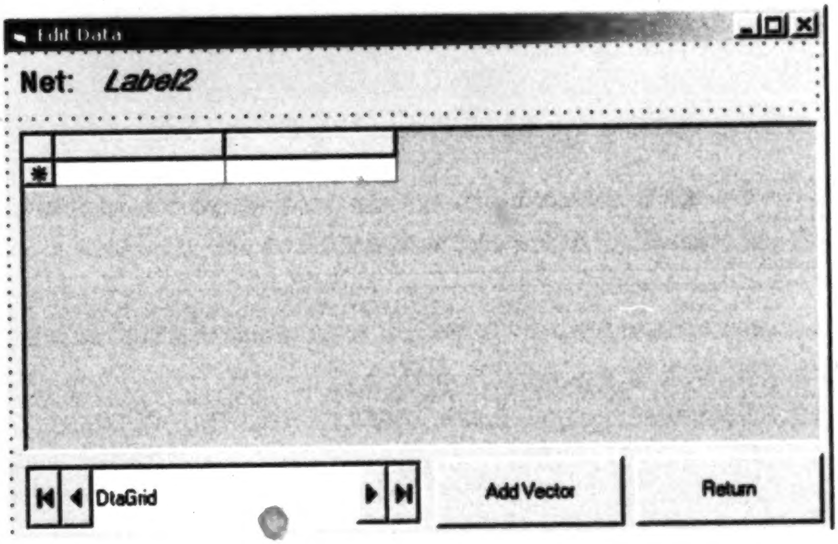

Edit Data
Net:
Label2
DtaGrid
Add Vector
Return

Figure A2-4. Neural Network Database Program Data Editing Form Design View

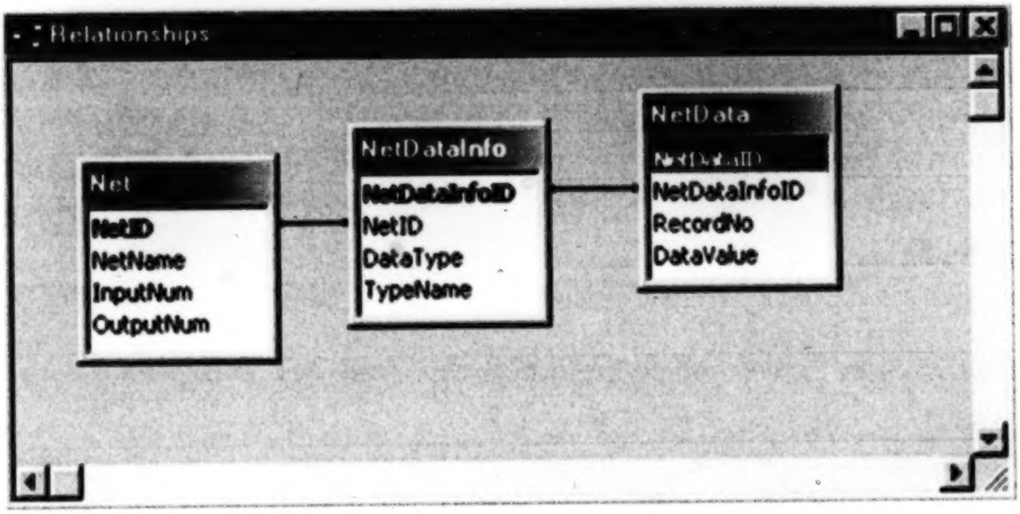

Relationships
Net
NetID
NetName
InputNum
OutputNum
NetDataInfo
NetDataInfoID
NetID
DataType
TypeName
NetData
NetDataID
NetDataInfoID
RecordNo
DataValue

Figure A2-5. Neural Network Database Relational Diagram

A2.3 Database Program Code

The following is the Visual Basic Code used to construct the Neural Network database program.

A2.3.1 Programming Code For The Main Form

```
Public CNetID As Integer
Public CName As String
Public CInputs As Integer
Public COutputs As Integer
```

```
Private Sub CmdAddNew_Click()
‘ add a new network to the database
    Dim NewName As String
    Dim NewInputs As Integer
    Dim NewOutputs As Integer
    Dim X As Integer
    Dim VarName As String

    ‘get net name and other information
    NewName = InputBox("Enter Net Name", "Net Definition")
    NewInputs = InputBox("Enter Number Of Input Variables For Net", "Net Definition")
    NewOutputs = InputBox("Enter Number Of Output Variables For Net", "Net Definition")
```

```
Dim dbs As Database

Dim Rst As Recordset


Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\Net.mdb")

Set Rst = dbs.TableDefs("Net").OpenRecordset


' add the network to the database

Rst.AddNew

   Rst!NetName = NewName

   Rst!InputNum = NewInputs

   Rst!OutputNum = NewOutputs

   CNetID = Rst!NetID

Rst.Update


CName = NewName

CInputs = NewInputs

COutputs = NewOutputs


'get variable information and add it to the database

Set Rst = dbs.TableDefs("NetDataInfo").OpenRecordset


For X = 1 To NewInputs
```

```
        VarName = InputBox("Name for INPUT" & X & " Variable", "Net Definition")
        Rst.AddNew
            Rst!NetID = CNetID
            Rst!DataType = "Input" & X
            Rst!TypeName = VarName
        Rst.Update
    Next X

    For X = 1 To NewOutputs
        VarName = InputBox("Name for OUTPUT" & X & " Variable", "Net Definition")
        Rst.AddNew
            Rst!NetID = CNetID
            Rst!DataType = "Output" & X
            Rst!TypeName = VarName
        Rst.Update
    Next X

    NetDB.Refresh

End Sub
```

```
Private Sub CmdEditData_Click()
'bring up the other form
```

```
   Form1.Visible = False
   FrmEditData.Show 1
End Sub
```

```
Private Sub CmdEnd_Click()
‘end the program
   End
End Sub
```

```
Private Sub NetGrid_RowColChange(LastRow As Variant, ByVal LastCol As Integer)
‘get information for next form
   NetGrid.Col = 0
   CNetID = NetGrid.Text

   NetGrid.Col = 1
   CName = NetGrid.Text

   NetGrid.Col = 2
   CInputs = NetGrid.Text

   NetGrid.Col = 3
   COutputs = NetGrid.Text
```

```
 NetGrid.Col = 0
End Sub
```

A2.3.2 Programming Code For The Data Editing Form

```
Private Sub CmdAddVector_Click()
'add a data vector to the database
  Dim dbs As Database
  Dim Rst As Recordset
  Dim Rst2 As Recordset
  Dim NextRecNo As Integer
  Dim X As Integer
  Dim VecVal As Double
  Dim NetInfoId As Integer

  Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\Net.mdb")
  Set Rst = dbs.TableDefs("NetData").OpenRecordset
  Set Rst2 = dbs.TableDefs("NetDataInfo").OpenRecordset

  NextRecNo = 0

  Rst.MoveFirst
  Do Until Rst.EOF
  If Rst!RecordNo > NextRecNo Then NextRecNo = Rst!RecordNo
  Rst.MoveNext
Loop
NextRecNo = NextRecNo + 1

For X = 1 To Form1.CInputs
  Rst2.MoveFirst
  Do Until Rst2.EOF
    If Rst2!DataType = "Input" & X Then
      If Rst2!NetID = Form1.CNetID Then
        NetInfoId = Rst2!NetDataInfoID
      End If
    End If
    Rst2.MoveNext
  Loop
  VecVal = InputBox("Enter the value for INPUT" & X, "Data Entry")
  Rst.AddNew
    Rst!NetDataInfoID = NetInfoId
    Rst!RecordNo = NextRecNo
    Rst!datavalue = VecVal
  Rst.Update
Next X
```

```
For X = 1 To Form1.COutputs
  Rst2.MoveFirst
  Do Until Rst2.EOF
    If Rst2!DataType = "Output" & X Then
      If Rst2!NetID = Form1.CNetID Then
        NetInfoId = Rst2!NetDataInfoID
      End If
    End If
    Rst2.MoveNext
  Loop
  VecVal = InputBox("Enter the value for OUTPUT" & X, "Data Entry")
  Rst.AddNew
    Rst!NetDataInfoID = NetInfoId
    Rst!RecordNo = NextRecNo
    Rst!datavalue = VecVal
  Rst.Update
Next X
DtaGrid.RecordSource = "Net"
DtaGrid.Refresh
GrdData.Refresh
ClearDB
MakeTablesForGrid
DtaGrid.RecordSource = "GridData"
```

```
  DtaGrid.Refresh
  GrdData.Refresh
End Sub
```

```
Private Sub CmdReturn_Click()
'return to the main form
   FrmEditData.Visible = False
   Form1.Show 1
End Sub
```

```
Private Sub Form_Activate()
'initial information for the form
   DtaGrid.RecordSource = "Net"
   DtaGrid.Refresh
   GrdData.Refresh
   ClearDB
   MakeTablesForGrid
   DtaGrid.RecordSource = "GridData"
   DtaGrid.Refresh
   GrdData.Refresh
   LblCName.Caption = Form1.CName
End Sub
```

```
Public Sub MakeTablesForGrid()
'get information for grid
   Dim dbs As Database
   Dim Rst As Recordset
   Dim X As Integer
   Dim SQLString As String
   Dim InputString As String
   Dim PlaceHolder As Integer
   Dim TotVaris As Integer
   TotVaris = Form1.CInputs + Form1.COutputs


   Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\Net.mdb")
   Set Rst = dbs.TableDefs("NetData").OpenRecordset
   For X = 1 To Form1.CInputs
      InputString = "'Input" & X & "'"
      SQLString = "SELECT NetDataInfo.NetID, NetData.RecordNo, NetData.DataValue,
NetDataInfo.DataType INTO " & X
      SQLString = SQLString + " FROM NetDataInfo INNER JOIN NetData ON
NetDataInfo.NetDataInfoID = NetData.NetDataInfoID "
      SQLString = SQLString + "WHERE NetDataInfo.DataType =" & InputString
      dbs.Execute SQLString
   Next X
```

```
  PlaceHolder = X
  For X = PlaceHolder To TotVaris
      InputString = "'Output" & (X - PlaceHolder + 1) & "'"
      SQLString = "SELECT NetDataInfo.NetID, NetData.RecordNo, NetData.DataValue,
NetDataInfo.DataType INTO " & X
      SQLString = SQLString + " FROM NetDataInfo INNER JOIN NetData ON
NetDataInfo.NetDataInfoID = NetData.NetDataInfoID "
      SQLString = SQLString + "WHERE NetDataInfo.DataType =" & InputString
      dbs.Execute SQLString
  Next X

  SQLString = "SELECT [1].[DataValue]"
  For X = 2 To TotVaris
      SQLString = SQLString + ", [" & X & "].[DataValue]"
  Next X
  SQLString = SQLString + " into GridData FROM "
  For X = 1 To (TotVaris - 1)
      SQLString = SQLString + "("
  Next X
  SQLString = SQLString + "1 INNER JOIN 2 ON [1].[RecordNo]=[2].[RecordNo] AND
[1].[NetID]=[2].[NetID])"
  For X = 2 To (TotVaris - 1)
```

```
    SQLString = SQLString + " INNER JOIN " & (X + 1) & " ON [" & X &
"].[RecordNo]=[" & (X + 1) & "].[RecordNo] AND [" & X & "].[NetID]=[" & (X + 1) &
"].[NetID])"
  Next X

  SQLString = SQLString + "WHERE [1].[NetID]=" & Form1.CNetID
   dbs.Execute SQLString
End Sub
```

```
Public Sub ClearDB()
'cleaning out the temp tables from the database
Dim dbs As Database
Dim tbl As TableDef
Dim X As Integer
Dim Total As Integer
Dim SQL As String

Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\Net.mdb")
Total = Form1.CInputs + Form1.COutputs

For X = 1 To Total
    For Each tbl In dbs.TableDefs
        SQL = "Drop Table " & X
        If tbl.Name = CStr(X) Then dbs.Execute SQL
    Next tbl
Next X
For Each tbl In dbs.TableDefs
        If tbl.Name = "GridData" Then dbs.Execute "Drop Table GridData"
Next tbl
dbs.Close
End Sub
```

APPENDIX 3 – Belief Network Information

A3.1 Overview

The belief networks constructed for the agents in this research was done through Microsoft Research's Bayesian Network Authoring and Evaluation Tool MSBNx (1.4). The following sections provide the information required to construct the networks developed for the examples presented in this thesis. In each section a figure will be presented to illustrate the constructs of the network. A "Belief Network Node Property Descriptions Table" will be presented that outlines the belief nodes property description value's that are required for the belief agent developed. Lastly the conditional property tables for each of the belief nodes are provided.

A3.2 Belief Network Construction Information

A3.2.1 Train Belief Network

The "train" belief network was designed for use by the agent in the tunnel construction simulation example presented in Chapter 3 and Appendix 4. The following information provides the necessary data to construct the belief network.

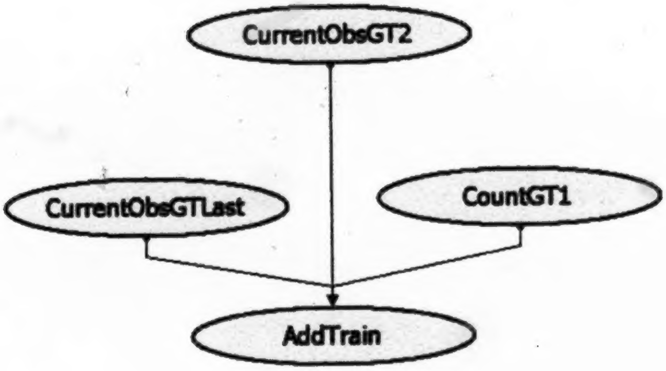

CurrentObsGT2
CurrentObsGTLast
CountGT1
AddTrain

Figure A3-1. Train Belief Network

Table A3-1. Belief Network Node Property Descriptions Table

Node	Property Description
CurrentObsGTLast	Input
CurrentObsGT2	Input
CountGT1	Input
AddTrain	Output

Table A3-2. CurrentObsGTLast Node Conditional Probability Table

CurrentObsGTLast	
Yes	0.5
No	0.5

Table A3-3. CurrentObsGT2 Node Conditional Probability Table

CurrentObsGT2	
Yes	0.5
No	0.5

Table A3-4. CountGT1 Node Conditional Probability Table

CountGT1	
Yes	0.5
No	0.5

Table A3-5. AddTrain Node Conditional Probability Table

			AddTrain	
CurrentObsGTLast	CurrentObsGT2	CountGT1	Yes	No
Yes	Yes	Yes	1.0	0.0
Yes	Yes	No	0.6	0.4
Yes	No	Yes	0.6	0.4
Yes	No	No	0.3	0.7
No	Yes	Yes	0.6	0.4
No	Yes	No	0.3	0.7
No	No	Yes	0.3	0.7
No	No	No	0.0	1.0

A3.2.2 Play Belief Network

The "play" belief network was designed for use by the agent in the truck queuing simulation example presented in Chapter 2 and Appendix 5. The following information provides the necessary data to construct the belief network.

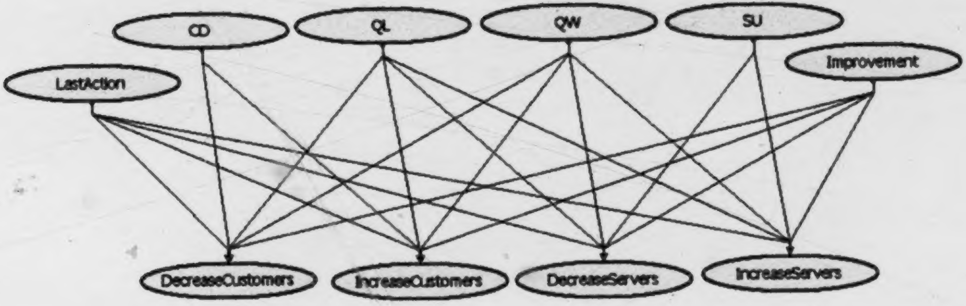

CD
QL
QW
SU
Improvement
LastAction
DecreaseCustomers
IncreaseCustomers
DecreaseServers
IncreaseServers

Figure A3-2. Play Belief Network

Table A3-6. Belief Network Node Property Descriptions Table

Node	Property Description
LastAction	Input
CD	Input
QL	Input
QW	Input
SU	Input
Improvement	Input
DecreaseCustomers	Output
IncreaseCustomers	Output
DecreaseServers	Output
IncreaseServers	Output

Table A3-7. LastAction Node Conditional Probability Table

LastAction	
DecreaseCustomers	0.25
IncreaseCustomers	0.25
DecreaseServers	0.25
IncreaseServers	0.25

Table A3-8. CD Node Conditional Probability Table

CD	
CD < CD Upper	0.5
CD > CD Upper	0.5

Table A3-9. QL Node Conditional Probability Table

QL	
QL Lower < QL < QL Upper	0.33
QL < QL Lower	0.33
QL > QL Upper	0.33

Table A3-10. QW Node Conditional Probability Table

QW	
QW Lower < QW < QW Upper	0.33
QW < QW Lower	0.33
QW > QW Upper	0.33

Table A3-11. SU Node Conditional Probability Table

SU	
SU Lower < SU < SU Upper	0.33
SU < SU Lower	0.33
SU > SU Upper	0.33

Table A3-12. Improvement Node Conditional Probability Table

Improvement	
Yes	0.5
No	0.5

Table A3-13. DecreaseCustomers Node Conditional Probability Table

					DecreaseCustomers	
Improvement	LastAction	CD	QL	QW	Yes	No
Yes	DecreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.90	0.10
Yes	DecreaseCustomers	CD < [illegible] Upp[illegible]	QL Lower < QL < QL Upper	QW < QW Lower	0.87	0.13
Yes	DecreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.93	0.07
Yes	DecreaseCustomers	CD < CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.85	0.15
Yes	DecreaseCustomers	CD < CD Upper	QL < QL Lower	QW < QW Lower	0.82	0.18
Yes	DecreaseCustomers	CD < CD Upper	QL < QL Lower	QW > QW Upper	0.88	0.12

					DecreaseCustomers	
Improvement	LastAction	CD	QL	QW	Yes	No
Yes	DecreaseCustomers	CD < CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.95	0.05
Yes	DecreaseCustomers	CD < CD Upper	QL > QL Upper	QW < QW Lower	0.92	0.08
Yes	DecreaseCustomers	CD < CD Upper	QL > QL Upper	QW > QW Upper	0.98	0.02
Yes	DecreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.70	0.30
Yes	DecreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.67	0.33
Yes	DecreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.73	0.27
Yes	DecreaseCustomers	CD > CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.65	0.35
Yes	DecreaseCustomers	CD > CD Upper	QL < QL Lower	QW < QW Lower	0.62	0.38
Yes	DecreaseCustomers	CD > CD Upper	QL < QL Lower	QW > QW Upper	0.68	0.32
Yes	DecreaseCustomers	CD > CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.75	0.25
Yes	DecreaseCustomers	CD > CD Upper	QL > QL Upper	QW < QW Lower	0.72	0.28
Yes	DecreaseCustomers	CD > CD Upper	QL > QL Upper	QW > QW Upper	0.78	0.22
Yes	IncreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.30	0.70
Yes	IncreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.27	0.73
Yes	IncreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.33	0.67
Yes	IncreaseCustomers	CD < CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.25	0.75
Yes	IncreaseCustomers	CD < CD Upper	QL < QL Lower	QW < QW Lower	0.23	0.77
Yes	IncreaseCustomers	CD < CD Upper	QL < QL Lower	QW > QW Upper	0.28	0.72
Yes	IncreaseCustomers	CD < CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.30	0.70
Yes	IncreaseCustomers	CD < CD Upper	QL > QL Upper	QW < QW Lower	0.27	0.73
Yes	IncreaseCustomers	CD < CD Upper	QL > QL Upper	QW > QW Upper	0.33	0.67
Yes	IncreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.10	0.90
Yes	IncreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.07	0.93
Yes	IncreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.13	0.87
Yes	IncreaseCustomers	CD > CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.05	0.95
Yes	IncreaseCustomers	CD > CD Upper	QL < QL Lower	QW < QW Lower	0.02	0.98
Yes	IncreaseCustomers	CD > CD Upper	QL < QL Lower	QW > QW Upper	0.08	0.92
Yes	IncreaseCustomers	CD > CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.15	0.85
Yes	IncreaseCustomers	CD > CD Upper	QL > QL Upper	QW < QW Lower	0.12	0.88
Yes	IncreaseCustomers	CD > CD Upper	QL > QL Upper	QW > QW Upper	0.18	0.82
Yes	DecreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.65	0.35
Yes	DecreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.62	0.38

Improvement	LastAction	CD	QL	QW	DecreaseCustomers	
					Yes	No
Yes	DecreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.68	0.32
Yes	DecreaseServers	CD < CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.60	0.40
Yes	DecreaseServers	CD < CD Upper	QL < QL Lower	QW < QW Lower	0.57	0.43
Yes	DecreaseServers	CD < CD Upper	QL < QL Lower	QW > QW Upper	0.63	0.37
Yes	DecreaseServers	CD < CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.67	0.33
Yes	DecreaseServers	CD < CD Upper	QL > QL Upper	QW < QW Lower	0.67	0.33
Yes	DecreaseServers	CD < CD Upper	QL > QL Upper	QW > QW Upper	0.73	0.27
Yes	DecreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.60	0.40
Yes	DecreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.57	0.43
Yes	DecreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.63	0.37
Yes	DecreaseServers	CD > CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.55	0.45
Yes	DecreaseServers	CD > CD Upper	QL < QL Lower	QW < QW Lower	0.57	0.43
Yes	DecreaseServers	CD > CD Upper	QL < QL Lower	QW > QW Upper	0.58	0.42
Yes	DecreaseServers	CD > CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.65	0.35
Yes	DecreaseServers	CD > CD Upper	QL > QL Upper	QW < QW Lower	0.62	0.38
Yes	DecreaseServers	CD > CD Upper	QL > QL Upper	QW > QW Upper	0.68	0.32
Yes	IncreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.40	0.60
Yes	IncreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.37	0.63
Yes	IncreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.43	0.57
Yes	IncreaseServers	CD < CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.35	0.65
Yes	IncreaseServers	CD < CD Upper	QL < QL Lower	QW < QW Lower	0.32	0.68
Yes	IncreaseServers	CD < CD Upper	QL < QL Lower	QW > QW Upper	0.38	0.62
Yes	IncreaseServers	CD < CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.45	0.55
Yes	IncreaseServers	CD < CD Upper	QL > QL Upper	QW < QW Lower	0.42	0.58
Yes	IncreaseServers	CD < CD Upper	QL > QL Upper	QW > QW Upper	0.48	0.52
Yes	IncreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.35	0.65
Yes	IncreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.32	0.68
Yes	IncreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.38	0.62
Yes	IncreaseServers	CD > CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.30	0.70
Yes	IncreaseServers	CD > CD Upper	QL < QL Lower	QW < QW Lower	0.27	0.73
Yes	IncreaseServers	CD > CD Upper	QL < QL Lower	QW > QW Upper	0.33	0.67
Yes	IncreaseServers	CD > CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.40	0.60

					DecreaseCustomers	
Improvement	LastAction	CD	QL	QW	Yes	No
Yes	IncreaseServers	CD > CD Upper	QL > QL Upper	QW < QW Lower	0.37	0.63
Yes	IncreaseServers	CD > CD Upper	QL > QL Upper	QW > QW Upper	0.43	0.57
No	DecreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.10	0.90
No	DecreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.07	0.93
No	DecreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.13	0.87
No	DecreaseCustomers	CD < CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.05	0.95
No	DecreaseCustomers	CD < CD Upper	QL < QL Lower	QW < QW Lower	0.02	0.98
No	DecreaseCustomers	CD < CD Upper	QL < QL Lower	QW > QW Upper	0.08	0.92
No	DecreaseCustomers	CD < CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.15	0.85
No	DecreaseCustomers	CD < CD Upper	QL > QL Upper	QW < QW Lower	0.12	0.88
No	DecreaseCustomers	CD < CD Upper	QL > QL Upper	QW > QW Upper	0.18	0.82
No	DecreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.30	0.70
No	DecreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.27	0.73
No	DecreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.33	0.67
No	DecreaseCustomers	CD > CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.25	0.75
No	DecreaseCustomers	CD > CD Upper	QL < QL Lower	QW < QW Lower	0.22	0.78
No	DecreaseCustomers	CD > CD Upper	QL < QL Lower	QW > QW Upper	0.28	0.72
No	DecreaseCustomers	CD > CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.35	0.65
No	DecreaseCustomers	CD > CD Upper	QL > QL Upper	QW < QW Lower	0.32	0.68
No	DecreaseCustomers	CD > CD Upper	QL > QL Upper	QW > QW Upper	0.38	0.62
No	IncreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.70	0.30
No	IncreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.67	0.33
No	IncreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.73	0.27
No	IncreaseCustomers	CD < CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.65	0.35
No	IncreaseCustomers	CD < CD Upper	QL < QL Lower	QW < QW Lower	0.62	0.38
No	IncreaseCustomers	CD < CD Upper	QL < QL Lower	QW > QW Upper	0.68	0.32
No	IncreaseCustomers	CD < CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.75	0.25
No	IncreaseCustomers	CD < CD Upper	QL > QL Upper	QW < QW Lower	0.72	0.28
No	IncreaseCustomers	CD < CD Upper	QL > QL Upper	QW > QW Upper	0.78	0.22
No	IncreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.90	0.10
No	IncreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.87	0.13
No	IncreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.93	0.07

					DecreaseCustomers	
Improvement	LastAction	CD	QL	QW	Yes	No
No	IncreaseCustomers	CD > CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.85	0.15
No	IncreaseCustomers	CD > CD Upper	QL < QL Lower	QW < QW Lower	0.82	0.18
No	IncreaseCustomers	CD > CD Upper	QL < QL Lower	QW > QW Upper	0.88	0.12
No	IncreaseCustomers	CD > CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.95	0.05
No	IncreaseCustomers	CD > CD Upper	QL > QL Upper	QW < QW Lower	0.92	0.08
No	IncreaseCustomers	CD > CD Upper	QL > QL Upper	QW > QW Upper	0.98	0.02
No	DecreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.35	0.65
No	DecreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.32	0.68
No	DecreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.38	0.62
No	DecreaseServers	CD < CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.30	0.70
No	DecreaseServers	CD < CD Upper	QL < QL Lower	QW < QW Lower	0.27	0.73
No	DecreaseServers	CD < CD Upper	QL < QL Lower	QW > QW Upper	0.33	0.67
No	DecreaseServers	CD < CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.40	0.60
No	DecreaseServers	CD < CD Upper	QL > QL Upper	QW < QW Lower	0.37	0.63
No	DecreaseServers	CD < CD Upper	QL > QL Upper	QW > QW Upper	0.43	0.57
No	DecreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.40	0.60
No	DecreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.37	0.63
No	DecreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.43	0.57
No	DecreaseServers	CD > CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.35	0.65
No	DecreaseServers	CD > CD Upper	QL < QL Lower	QW < QW Lower	0.32	0.68
No	DecreaseServers	CD > CD Upper	QL < QL Lower	QW > QW Upper	0.38	0.62
No	DecreaseServers	CD > CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.45	0.55
No	DecreaseServers	CD > CD Upper	QL > QL Upper	QW < QW Lower	0.42	0.58
No	DecreaseServers	CD > CD Upper	QL > QL Upper	QW > QW Upper	0.48	0.52
No	IncreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.60	0.40
No	IncreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.57	0.43
No	IncreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.63	0.37
No	IncreaseServers	CD < CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.55	0.45
No	IncreaseServers	CD < CD Upper	QL < QL Lower	QW < QW Lower	0.52	0.48
No	IncreaseServers	CD < CD Upper	QL < QL Lower	QW > QW Upper	0.58	0.42
No	IncreaseServers	CD < CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.65	0.35
No	IncreaseServers	CD < CD Upper	QL > QL Upper	QW < QW Lower	0.62	0.38

					DecreaseCustomers	
Improvement	LastAction	CD	QL	QW	Yes	No
No	IncreaseServers	CD < CD Upper	QL > QL Upper	QW > QW Upper	0.68	0.32
No	IncreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.65	0.35
No	IncreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.62	0.38
No	IncreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.68	0.32
No	IncreaseServers	CD > CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.70	0.30
No	IncreaseServers	CD > CD Upper	QL < QL Lower	QW < QW Lower	0.67	0.33
No	IncreaseServers	CD > CD Upper	QL < QL Lower	QW > QW Upper	0.73	0.27
No	IncreaseServers	CD > CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.60	0.40
No	IncreaseServers	CD > CD Upper	QL > QL Upper	QW < QW Lower	0.57	0.43
No	IncreaseServers	CD > CD Upper	QL > QL Upper	QW > QW Upper	0.63	0.37

Table A3-14. IncreaseCustomers Node Conditional Probability Table

					IncreaseCustomers	
Improvement	LastAction	CD	QL	QW	Yes	No
Yes	DecreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.30	0.70
Yes	DecreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.33	0.67
Yes	DecreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.27	0.73
Yes	DecreaseCustomers	CD < CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.35	0.65
Yes	DecreaseCustomers	CD < CD Upper	QL < QL Lower	QW < QW Lower	0.38	0.62
Yes	DecreaseCustomers	CD < CD Upper	QL < QL Lower	QW > QW Upper	0.32	0.68
Yes	DecreaseCustomers	CD < CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.25	0.75
Yes	DecreaseCustomers	CD < CD Upper	QL > QL Upper	QW < QW Lower	0.28	0.72
Yes	DecreaseCustomers	CD < CD Upper	QL > QL Upper	QW > QW Upper	0.22	0.78
Yes	DecreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.10	0.90
Yes	DecreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.13	0.87
Yes	DecreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.07	0.93
Yes	DecreaseCustomers	CD > CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.15	0.85
Yes	DecreaseCustomers	CD > CD Upper	QL < QL Lower	QW < QW Lower	0.18	0.82
Yes	DecreaseCustomers	CD > CD Upper	QL < QL Lower	QW > QW Upper	0.12	0.88
Yes	DecreaseCustomers	CD > CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.05	0.95
Yes	DecreaseCustomers	CD > CD Upper	QL > QL Upper	QW < QW Lower	0.08	0.92
Yes	DecreaseCustomers	CD > CD Upper	QL > QL Upper	QW > QW Upper	0.02	0.98

					IncreaseCustomers	
Improvement	LastAction	CD	QL	QW	Yes	No
Yes	IncreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.90	0.10
Yes	IncreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.98	0.02
Yes	IncreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.92	0.08
Yes	IncreaseCustomers	CD < CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.95	0.05
Yes	IncreaseCustomers	CD < CD Upper	QL < QL Lower	QW < QW Lower	0.98	0.02
Yes	IncreaseCustomers	CD < CD Upper	QL < QL Lower	QW > QW Upper	0.92	0.08
Yes	IncreaseCustomers	CD < CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.85	0.15
Yes	IncreaseCustomers	CD < CD Upper	QL > QL Upper	QW < QW Lower	0.88	0.12
Yes	IncreaseCustomers	CD < CD Upper	QL > QL Upper	QW > QW Upper	0.82	0.18
Yes	IncreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.70	0.30
Yes	IncreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.73	0.27
Yes	IncreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.67	0.33
Yes	IncreaseCustomers	CD > CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.75	0.25
Yes	IncreaseCustomers	CD > CD Upper	QL < QL Lower	QW < QW Lower	0.78	0.22
Yes	IncreaseCustomers	CD > CD Upper	QL < QL Lower	QW > QW Upper	0.72	0.28
Yes	IncreaseCustomers	CD > CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.65	0.35
Yes	IncreaseCustomers	CD > CD Upper	QL > QL Upper	QW < QW Lower	0.68	0.32
Yes	IncreaseCustomers	CD > CD Upper	QL > QL Upper	QW > QW Upper	0.62	0.38
Yes	DecreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.40	0.60
Yes	DecreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.43	0.57
Yes	DecreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.37	0.63
Yes	DecreaseServers	CD < CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.45	0.55
Yes	DecreaseServers	CD < CD Upper	QL < QL Lower	QW < QW Lower	0.48	0.52
Yes	DecreaseServers	CD < CD Upper	QL < QL Lower	QW > QW Upper	0.42	0.58
Yes	DecreaseServers	CD < CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.35	0.65
Yes	DecreaseServers	CD < CD Upper	QL > QL Upper	QW < QW Lower	0.38	0.62
Yes	DecreaseServers	CD < CD Upper	QL > QL Upper	QW > QW Upper	0.32	0.68
Yes	DecreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.35	0.65
Yes	DecreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.38	0.62
Yes	DecreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.32	0.68
Yes	DecreaseServers	CD > CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.40	0.60
Yes	DecreaseServers	CD > CD Upper	QL < QL Lower	QW < QW Lower	0.43	0.57

					IncreaseCustomers	
Improvement	LastAction	CD	QL	QW	Yes	No
Yes	DecreaseServers	CD > CD Upper	QL < QL Lower	QW > QW Upper	0.37	0.63
Yes	DecreaseServers	CD > CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.30	0.70
Yes	DecreaseServers	CD > CD Upper	QL > QL Upper	QW < QW Lower	0.33	0.67
Yes	DecreaseServers	CD > CD Upper	QL > QL Upper	QW > QW Upper	0.27	0.73
Yes	IncreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.65	0.35
Yes	IncreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.68	0.32
Yes	IncreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.62	0.38
Yes	IncreaseServers	CD < CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.70	0.30
Yes	IncreaseServers	CD < CD Upper	QL < QL Lower	QW < QW Lower	0.73	0.27
Yes	IncreaseServers	CD < CD Upper	QL < QL Lower	QW > QW Upper	0.67	0.33
Yes	IncreaseServers	CD < CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.60	0.40
Yes	IncreaseServers	CD < CD Upper	QL > QL Upper	QW < QW Lower	0.63	0.37
Yes	IncreaseServers	CD < CD Upper	QL > QL Upper	QW > QW Upper	0.57	0.43
Yes	IncreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.60	0.40
Yes	IncreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.63	0.37
Yes	IncreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.57	0.43
Yes	IncreaseServers	CD > CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.65	0.35
Yes	IncreaseServers	CD > CD Upper	QL < QL Lower	QW < QW Lower	0.68	0.32
Yes	IncreaseServers	CD > CD Upper	QL < QL Lower	QW > QW Upper	0.62	0.38
Yes	IncreaseServers	CD > CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.55	0.45
Yes	IncreaseServers	CD > CD Upper	QL > QL Upper	QW < QW Lower	0.58	0.42
Yes	IncreaseServers	CD > CD Upper	QL > QL Upper	QW > QW Upper	0.52	0.48
No	DecreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.90	0.10
No	DecreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.93	0.07
No	DecreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.87	0.13
No	DecreaseCustomers	CD < CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.95	0.05
No	DecreaseCustomers	CD < CD Upper	QL < QL Lower	QW < QW Lower	0.98	0.02
No	DecreaseCustomers	CD < CD Upper	QL < QL Lower	QW > QW Upper	0.92	0.08
No	DecreaseCustomers	CD < CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.85	0.15
No	DecreaseCustomers	CD < CD Upper	QL > QL Upper	QW < QW Lower	0.88	0.12
No	DecreaseCustomers	CD < CD Upper	QL > QL Upper	QW > QW Upper	0.82	0.18
No	DecreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.70	0.30

Improvement	LastAction	CD	QL	QW	IncreaseCustomers	
					Yes	No
No	DecreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.73	0.27
No	DecreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.67	0.33
No	DecreaseCustomers	CD > CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.75	0.25
No	DecreaseCustomers	CD > CD Upper	QL < QL Lower	QW < QW Lower	0.78	0.22
No	DecreaseCustomers	CD > CD Upper	QL < QL Lower	QW > QW Upper	0.72	0.28
No	DecreaseCustomers	CD > CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.65	0.35
No	DecreaseCustomers	CD > CD Upper	QL > QL Upper	QW < QW Lower	0.68	0.32
No	DecreaseCustomers	CD > CD Upper	QL > QL Upper	QW > QW Upper	0.62	0.38
No	IncreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.30	0.70
No	IncreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.33	0.67
No	IncreaseCustomers	CD < CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.27	0.73
No	IncreaseCustomers	CD < CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.35	0.65
No	IncreaseCustomers	CD < CD Upper	QL < QL Lower	QW < QW Lower	0.38	0.62
No	IncreaseCustomers	CD < CD Upper	QL < QL Lower	QW > QW Upper	0.32	0.68
No	IncreaseCustomers	CD < CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.25	0.75
No	IncreaseCustomers	CD < CD Upper	QL > QL Upper	QW < QW Lower	0.28	0.72
No	IncreaseCustomers	CD < CD Upper	QL > QL Upper	QW > QW Upper	0.22	0.78
No	IncreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.10	0.90
No	IncreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.13	0.87
No	IncreaseCustomers	CD > CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.07	0.93
No	IncreaseCustomers	CD > CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.15	0.85
No	IncreaseCustomers	CD > CD Upper	QL < QL Lower	QW < QW Lower	0.18	0.82
No	IncreaseCustomers	CD > CD Upper	QL < QL Lower	QW > QW Upper	0.12	0.88
No	IncreaseCustomers	CD > CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.05	0.95
No	IncreaseCustomers	CD > CD Upper	QL > QL Upper	QW < QW Lower	0.80	0.20
No	IncreaseCustomers	CD > CD Upper	QL > QL Upper	QW > QW Upper	0.20	0.80
No	DecreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.65	0.35
No	DecreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.68	0.32
No	DecreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.62	0.38
No	DecreaseServers	CD < CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.70	0.30
No	DecreaseServers	CD < CD Upper	QL < QL Lower	QW < QW Lower	0.73	0.27
No	DecreaseServers	CD < CD Upper	QL < QL Lower	QW > QW Upper	0.67	0.33

					IncreaseCustomers	
Improvement	LastAction	CD	QL	QW	Yes	No
No	DecreaseServers	CD < CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.60	0.40
No	DecreaseServers	CD < CD Upper	QL > QL Upper	QW < QW Lower	0.63	0.37
No	DecreaseServers	CD < CD Upper	QL > QL Upper	QW > QW Upper	0.57	0.43
No	DecreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.60	0.40
No	DecreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.63	0.37
No	DecreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.57	0.43
No	DecreaseServers	CD > CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.65	0.35
No	DecreaseServers	CD > CD Upper	QL < QL Lower	QW < QW Lower	0.68	0.32
No	DecreaseServers	CD > CD Upper	QL < QL Lower	QW > QW Upper	0.62	0.38
No	DecreaseServers	CD > CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.55	0.45
No	DecreaseServers	CD > CD Upper	QL > QL Upper	QW < QW Lower	0.58	0.42
No	DecreaseServers	CD > CD Upper	QL > QL Upper	QW > QW Upper	0.62	0.38
No	IncreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.40	0.60
No	IncreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.43	0.57
No	IncreaseServers	CD < CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.37	0.63
No	IncreaseServers	CD < CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.45	0.55
No	IncreaseServers	CD < CD Upper	QL < QL Lower	QW < QW Lower	0.48	0.52
No	IncreaseServers	CD < CD Upper	QL < QL Lower	QW > QW Upper	0.42	0.58
No	IncreaseServers	CD < CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.35	0.65
No	IncreaseServers	CD < CD Upper	QL > QL Upper	QW < QW Lower	0.38	0.62
No	IncreaseServers	CD < CD Upper	QL > QL Upper	QW > QW Upper	0.32	0.68
No	IncreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.35	0.65
No	IncreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.38	0.62
No	IncreaseServers	CD > CD Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.32	0.68
No	IncreaseServers	CD > CD Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.40	0.60
No	IncreaseServers	CD > CD Upper	QL < QL Lower	QW < QW Lower	0.43	0.57
No	IncreaseServers	CD > CD Upper	QL < QL Lower	QW > QW Upper	0.37	0.63
No	IncreaseServers	CD > CD Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.30	0.70
No	IncreaseServers	CD > CD Upper	QL > QL Upper	QW < QW Lower	0.33	0.67
No	IncreaseServers	CD > CD Upper	QL > QL Upper	QW > QW Upper	0.30	0.70

Table A3-15. DecreaseServers Node Conditional Probability Table

					DecreaseServers	
Improvement	LastAction	SU	QL	QW	Yes	No
Yes	DecreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.40	0.60
Yes	DecreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.43	0.57
Yes	DecreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.37	0.63
Yes	DecreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.45	0.55
Yes	DecreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW < QW Lower	0.48	0.52
Yes	DecreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW > QW Upper	0.42	0.58
Yes	DecreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.35	0.65
Yes	DecreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW < QW Lower	0.38	0.62
Yes	DecreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW > QW Upper	0.32	0.68
Yes	DecreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.30	0.70
Yes	DecreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW < QW Lower	0.33	0.67
Yes	DecreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW > QW Upper	0.27	0.73
Yes	DecreaseCustomers	SU < SU Lower	QL < QL Lower	QW Lower < QW < QW Upper	0.35	0.65
Yes	DecreaseCustomers	SU < SU Lower	QL < QL Lower	QW < QW Lower	0.38	0.62
Yes	DecreaseCustomers	SU < SU Lower	QL < QL Lower	QW > QW Upper	0.32	0.68
Yes	DecreaseCustomers	SU < SU Lower	QL > QL Upper	QW Lower < QW < QW Upper	0.25	0.75
Yes	DecreaseCustomers	SU < SU Lower	QL > QL Upper	QW < QW Lower	0.28	0.72
Yes	DecreaseCustomers	SU < SU Lower	QL > QL Upper	QW > QW Upper	0.22	0.78
Yes	DecreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.50	0.50
Yes	DecreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.53	0.47
Yes	DecreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.47	0.53
Yes	DecreaseCustomers	SU > SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.55	0.45
Yes	DecreaseCustomers	SU > SU Upper	QL < QL Lower	QW < QW Lower	0.58	0.42
Yes	DecreaseCustomers	SU > SU Upper	QL < QL Lower	QW > QW Upper	0.52	0.48
Yes	DecreaseCustomers	SU > SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.45	0.55
Yes	DecreaseCustomers	SU > SU Upper	QL > QL Upper	QW < QW Lower	0.48	0.52
Yes	DecreaseCustomers	SU > SU Upper	QL > QL Upper	QW > QW Upper	0.42	0.58
Yes	IncreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.60	0.40
Yes	IncreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.63	0.37
Yes	IncreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.57	0.43
Yes	IncreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.65	0.35
Yes	IncreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW < QW Lower	0.68	0.32
Yes	IncreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW > QW Upper	0.62	0.38

					DecreaseServers	
Improvement	LastAction	SU	QL	QW	Yes	No
Yes	IncreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.55	0.45
Yes	IncreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW < QW Lower	0.58	0.42
Yes	IncreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW > QW Upper	0.52	0.48
Yes	IncreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.50	0.50
Yes	IncreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW < QW Lower	0.53	0.47
Yes	IncreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW > QW Upper	0.47	0.53
Yes	IncreaseCustomers	SU < SU Lower	QL < QL Lower	QW Lower < QW < QW Upper	0.55	0.45
Yes	IncreaseCustomers	SU < SU Lower	QL < QL Lower	QW < QW Lower	0.58	0.42
Yes	IncreaseCustomers	SU < SU Lower	QL < QL Lower	QW > QW Upper	0.52	0.48
Yes	IncreaseCustomers	SU < SU Lower	QL > QL Upper	QW Lower < QW < QW Upper	0.50	0.50
Yes	IncreaseCustomers	SU < SU Lower	QL > QL Upper	QW < QW Lower	0.53	0.47
Yes	IncreaseCustomers	SU < SU Lower	QL > QL Upper	QW > QW Upper	0.47	0.53
Yes	IncreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.70	0.30
Yes	IncreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.73	0.27
Yes	IncreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.67	0.33
Yes	IncreaseCustomers	SU > SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.75	0.25
Yes	IncreaseCustomers	SU > SU Upper	QL < QL Lower	QW < QW Lower	0.78	0.22
Yes	IncreaseCustomers	SU > SU Upper	QL < QL Lower	QW > QW Upper	0.72	0.28
Yes	IncreaseCustomers	SU > SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.65	0.35
Yes	IncreaseCustomers	SU > SU Upper	QL > QL Upper	QW < QW Lower	0.68	0.32
Yes	IncreaseCustomers	SU > SU Upper	QL > QL Upper	QW > QW Upper	0.62	0.38
Yes	DecreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.90	0.10
Yes	DecreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.93	0.07
Yes	DecreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.87	0.13
Yes	DecreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.95	0.05
Yes	DecreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW < QW Lower	0.98	0.02
Yes	DecreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW > QW Upper	0.92	0.08
Yes	DecreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.85	0.15
Yes	DecreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW < QW Lower	0.88	0.12
Yes	DecreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW > QW Upper	0.82	0.18
Yes	DecreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.80	0.20
Yes	DecreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW < QW Lower	0.83	0.17
Yes	DecreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW > QW Upper	0.77	0.23
Yes	DecreaseServers	SU < SU Lower	QL < QL Lower	QW Lower < QW < QW Upper	0.85	0.15
Yes	DecreaseServers	SU < SU Lower	QL < QL Lower	QW < QW Lower	0.88	0.12

					DecreaseServers	
Improvement	LastAction	SU	QL	QW	Yes	No
Yes	DecreaseServers	SU < SU Lower	QL < QL Lower	QW > QW Upper	0.82	0.18
Yes	DecreaseServers	SU < SU Lower	QL > QL Upper	QW Lower < QW < QW Upper	0.75	0.25
Yes	DecreaseServers	SU < SU Lower	QL > QL Upper	QW < QW Lower	0.78	0.22
Yes	DecreaseServers	SU < SU Lower	QL > QL Upper	QW > QW Upper	0.72	0.28
Yes	DecreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.95	0.05
Yes	DecreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.98	0.02
Yes	DecreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.92	0.08
Yes	DecreaseServers	SU > SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.98	0.02
Yes	DecreaseServers	SU > SU Upper	QL < QL Lower	QW < QW Lower	1.00	0.00
Yes	DecreaseServers	SU > SU Upper	QL < QL Lower	QW > QW Upper	0.95	0.05
Yes	DecreaseServers	SU > SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.90	0.10
Yes	DecreaseServers	SU > SU Upper	QL > QL Upper	QW < QW Lower	0.93	0.07
Yes	DecreaseServers	SU > SU Upper	QL > QL Upper	QW > QW Upper	0.87	0.13
Yes	IncreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.10	0.90
Yes	IncreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.13	0.87
Yes	IncreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.07	0.93
Yes	IncreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.15	0.85
Yes	IncreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW < QW Lower	0.18	0.82
Yes	IncreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW > QW Upper	0.12	0.88
Yes	IncreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.50	0.50
Yes	IncreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW < QW Lower	0.53	0.47
Yes	IncreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW > QW Upper	0.47	0.53
Yes	IncreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.05	0.95
Yes	IncreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW < QW Lower	0.08	0.92
Yes	IncreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW > QW Upper	0.02	0.98
Yes	IncreaseServers	SU < SU Lower	QL < QL Lower	QW Lower < QW < QW Upper	0.98	0.02
Yes	IncreaseServers	SU < SU Lower	QL < QL Lower	QW < QW Lower	1.00	0.00
Yes	IncreaseServers	SU < SU Lower	QL < QL Lower	QW > QW Upper	0.95	0.05
Yes	IncreaseServers	SU < SU Lower	QL > QL Upper	QW Lower < QW < QW Upper	0.30	0.70
Yes	IncreaseServers	SU < SU Lower	QL > QL Upper	QW < QW Lower	0.33	0.67
Yes	IncreaseServers	SU < SU Lower	QL > QL Upper	QW > QW Upper	0.27	0.73
Yes	IncreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.20	0.80
Yes	IncreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.23	0.77
Yes	IncreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.17	0.83
Yes	IncreaseServers	SU > SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.25	0.75
Yes	IncreaseServers	SU > SU Upper	QL < QL Lower	QW < QW Lower	0.28	0.72

					DecreaseServers	
Improvement	LastAction	SU	QL	QW	Yes	No
Yes	IncreaseServers	SU > SU Upper	QL < QL Lower	QW > QW Upper	0.22	0.78
Yes	IncreaseServers	SU > SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.15	0.85
Yes	IncreaseServers	SU > SU Upper	QL > QL Upper	QW < QW Lower	0.18	0.82
Yes	IncreaseServers	SU > SU Upper	QL > QL Upper	QW > QW Upper	0.12	0.88
No	DecreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.60	0.40
No	DecreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.63	0.37
No	DecreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.57	0.43
No	DecreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.65	0.35
No	DecreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW < QW Lower	0.68	0.32
No	DecreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW > QW Upper	0.62	0.38
No	DecreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.55	0.45
No	DecreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW < QW Lower	0.58	0.42
No	DecreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW > QW Upper	0.52	0.48
No	DecreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.50	0.50
No	DecreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW < QW Lower	0.47	0.53
No	DecreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW > QW Upper	0.53	0.47
No	DecreaseCustomers	SU < SU Lower	QL < QL Lower	QW Lower < QW < QW Upper	0.55	0.45
No	DecreaseCustomers	SU < SU Lower	QL < QL Lower	QW < QW Lower	0.58	0.42
No	DecreaseCustomers	SU < SU Lower	QL < QL Lower	QW > QW Upper	0.52	0.48
No	DecreaseCustomers	SU < SU Lower	QL > QL Upper	QW Lower < QW < QW Upper	0.45	0.55
No	DecreaseCustomers	SU < SU Lower	QL > QL Upper	QW < QW Lower	0.48	0.52
No	DecreaseCustomers	SU < SU Lower	QL > QL Upper	QW > QW Upper	0.42	0.58
No	DecreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.70	0.30
No	DecreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.73	0.27
No	DecreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.67	0.33
No	DecreaseCustomers	SU > SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.75	0.25
No	DecreaseCustomers	SU > SU Upper	QL < QL Lower	QW < QW Lower	0.78	0.22
No	DecreaseCustomers	SU > SU Upper	QL < QL Lower	QW > QW Upper	0.72	0.28
No	DecreaseCustomers	SU > SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.65	0.35
No	DecreaseCustomers	SU > SU Upper	QL > QL Upper	QW < QW Lower	0.68	0.32
No	DecreaseCustomers	SU > SU Upper	QL > QL Upper	QW > QW Upper	0.62	0.38
No	IncreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.40	0.60
No	IncreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.43	0.57
No	IncreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.37	0.63
No	IncreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.45	0.55
No	IncreaseCustomers	SU Lower < SU	QL < QL Lower	QW < QW Lower	0.48	0.52

					DecreaseServers	
Improvement	LastAction	SU	QL	QW	Yes	No
		< SU Upper				
No	IncreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW > QW Upper	0.42	0.58
No	IncreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.35	0.65
No	IncreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW < QW Lower	0.38	0.62
No	IncreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW > QW Upper	0.32	0.68
No	IncreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.30	0.70
No	IncreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW < QW Lower	0.33	0.67
No	IncreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW > QW Upper	0.27	0.73
No	IncreaseCustomers	SU < SU Lower	QL < QL Lower	QW Lower < QW < QW Upper	0.35	0.65
No	IncreaseCustomers	SU < SU Lower	QL < QL Lower	QW < QW Lower	0.38	0.62
No	IncreaseCustomers	SU < SU Lower	QL < QL Lower	QW > QW Upper	0.32	0.68
No	IncreaseCustomers	SU < SU Lower	QL > QL Upper	QW Lower < QW < QW Upper	0.25	0.75
No	IncreaseCustomers	SU < SU Lower	QL > QL Upper	QW < QW Lower	0.27	0.73
No	IncreaseCustomers	SU < SU Lower	QL > QL Upper	QW > QW Upper	0.22	0.78
No	IncreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.50	0.50
No	IncreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.53	0.47
No	IncreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.47	0.53
No	IncreaseCustomers	SU > SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.55	0.45
No	IncreaseCustomers	SU > SU Upper	QL < QL Lower	QW < QW Lower	0.58	0.42
No	IncreaseCustomers	SU > SU Upper	QL < QL Lower	QW > QW Upper	0.52	0.48
No	IncreaseCustomers	SU > SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.45	0.55
No	IncreaseCustomers	SU > SU Upper	QL > QL Upper	QW < QW Lower	0.48	0.52
No	IncreaseCustomers	SU > SU Upper	QL > QL Upper	QW > QW Upper	0.42	0.58
No	DecreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.10	0.90
No	DecreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.13	0.87
No	DecreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.07	0.93
No	DecreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.15	0.85
No	DecreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW < QW Lower	0.18	0.82
No	DecreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW > QW Upper	0.12	0.88
No	DecreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.05	0.95
No	DecreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW < QW Lower	0.08	0.92
No	DecreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW > QW Upper	0.02	0.98
No	DecreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.05	0.95
No	DecreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW < QW Lower	0.08	0.92
No	DecreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW > QW Upper	0.02	0.98

					DecreaseServers	
Improvement	LastAction	SU	QL	QW	Yes	No
No	DecreaseServers	SU < SU Lower	QL < QL Lower	QW Lower < QW < QW Upper	0.10	0.90
No	DecreaseServers	SU < SU Lower	QL < QL Lower	QW < QW Lower	0.13	0.87
No	DecreaseServers	SU < SU Lower	QL < QL Lower	QW > QW Upper	0.07	0.93
No	DecreaseServers	SU < SU Lower	QL > QL Upper	QW Lower < QW < QW Upper	0.03	0.97
No	DecreaseServers	SU < SU Lower	QL > QL Upper	QW < QW Lower	0.08	0.92
No	DecreaseServers	SU < SU Lower	QL > QL Upper	QW > QW Upper	0.00	1.00
No	DecreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.20	0.80
No	DecreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.23	0.77
No	DecreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.17	0.83
No	DecreaseServers	SU > SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.25	0.75
No	DecreaseServers	SU > SU Upper	QL < QL Lower	QW < QW Lower	0.28	0.72
No	DecreaseServers	SU > SU Upper	QL < QL Lower	QW > QW Upper	0.22	0.78
No	DecreaseServers	SU > SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.15	0.85
No	DecreaseServers	SU > SU Upper	QL > QL Upper	QW < QW Lower	0.18	0.82
No	DecreaseServers	SU > SU Upper	QL > QL Upper	QW > QW Upper	0.12	0.88
No	IncreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.40	0.60
No	IncreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.43	0.57
No	IncreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.37	0.63
No	IncreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.45	0.55
No	IncreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW < QW Lower	0.48	0.52
No	IncreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW > QW Upper	0.42	0.58
No	IncreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.35	0.65
No	IncreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW < QW Lower	0.38	0.62
No	IncreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW > QW Upper	0.32	0.68
No	IncreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.30	0.70
No	IncreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW < QW Lower	0.33	0.67
No	IncreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW > QW Upper	0.27	0.73
No	IncreaseServers	SU < SU Lower	QL < QL Lower	QW Lower < QW < QW Upper	0.35	0.65
No	IncreaseServers	SU < SU Lower	QL < QL Lower	QW < QW Lower	0.38	0.62
No	IncreaseServers	SU < SU Lower	QL < QL Lower	QW > QW Upper	0.32	0.68
No	IncreaseServers	SU < SU Lower	QL > QL Upper	QW Lower < QW < QW Upper	0.25	0.75
No	IncreaseServers	SU < SU Lower	QL > QL Upper	QW < QW Lower	0.28	0.72
No	IncreaseServers	SU < SU Lower	QL > QL Upper	QW > QW Upper	0.22	0.78
No	IncreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.50	0.50
No	IncreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.53	0.47
No	IncreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.47	0.53

					DecreaseServers	
Improvement	LastAction	SU	QL	QW	Yes	No
No	IncreaseServers	SU > SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.55	0.45
No	IncreaseServers	SU > SU Upper	QL < QL Lower	QW < QW Lower	0.58	0.42
No	IncreaseServers	SU > SU Upper	QL < QL Lower	QW > QW Upper	0.52	0.48
No	IncreaseServers	SU > SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.45	0.55
No	IncreaseServers	SU > SU Upper	QL > QL Upper	QW < QW Lower	0.48	0.52
No	IncreaseServers	SU > SU Upper	QL > QL Upper	QW > QW Upper	0.42	0.58

Table A3-16. IncreaseServers Node Conditional Probability Table

					IncreaseServers	
Improvement	LastAction	SU	QL	QW	Yes	No
Yes	DecreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.60	0.40
Yes	DecreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.57	0.43
Yes	DecreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.63	0.37
Yes	DecreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.57	0.43
Yes	DecreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW < QW Lower	0.54	0.46
Yes	DecreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW > QW Upper	0.60	0.40
Yes	DecreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.63	0.37
Yes	DecreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW < QW Lower	0.60	0.40
Yes	DecreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW > QW Upper	0.66	0.34
Yes	DecreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.70	0.30
Yes	DecreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW < QW Lower	0.67	0.33
Yes	DecreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW > QW Upper	0.73	0.27
Yes	DecreaseCustomers	SU < SU Lower	QL < QL Lower	QW Lower < QW < QW Upper	0.67	0.33
Yes	DecreaseCustomers	SU < SU Lower	QL < QL Lower	QW < QW Lower	0.64	0.36
Yes	DecreaseCustomers	SU < SU Lower	QL < QL Lower	QW > QW Upper	0.70	0.30
Yes	DecreaseCustomers	SU < SU Lower	QL > QL Upper	QW Lower < QW < QW Upper	0.73	0.27
Yes	DecreaseCustomers	SU < SU Lower	QL > QL Upper	QW < QW Lower	0.70	0.30
Yes	DecreaseCustomers	SU < SU Lower	QL > QL Upper	QW > QW Upper	0.76	0.24
Yes	DecreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.50	0.50
Yes	DecreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.47	0.53
Yes	DecreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.53	0.47
Yes	DecreaseCustomers	SU > SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.47	0.53
Yes	DecreaseCustomers	SU > SU Upper	QL < QL Lower	QW < QW Lower	0.44	0.56

					IncreaseServers	
Improvement	LastAction	SU	QL	QW	Yes	No
Yes	DecreaseCustomers	SU > SU Upper	QL < QL Lower	QW > QW Upper	0.50	0.50
Yes	DecreaseCustomers	SU > SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.53	0.47
Yes	DecreaseCustomers	SU > SU Upper	QL > QL Upper	QW < QW Lower	0.50	0.50
Yes	DecreaseCustomers	SU > SU Upper	QL > QL Upper	QW > QW Upper	0.56	0.44
Yes	IncreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.40	0.60
Yes	IncreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.37	0.63
Yes	IncreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.43	0.57
Yes	IncreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.37	0.63
Yes	IncreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW < QW Lower	0.34	0.66
Yes	IncreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW > QW Upper	0.40	0.60
Yes	IncreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.43	0.57
Yes	IncreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW < QW Lower	0.40	0.60
Yes	IncreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW > QW Upper	0.46	0.54
Yes	IncreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.50	0.50
Yes	IncreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW < QW Lower	0.47	0.53
Yes	IncreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW > QW Upper	0.53	0.47
Yes	IncreaseCustomers	SU < SU Lower	QL < QL Lower	QW Lower < QW < QW Upper	0.47	0.53
Yes	IncreaseCustomers	SU < SU Lower	QL < QL Lower	QW < QW Lower	0.44	0.56
Yes	IncreaseCustomers	SU < SU Lower	QL < QL Lower	QW > QW Upper	0.50	0.50
Yes	IncreaseCustomers	SU < SU Lower	QL > QL Upper	QW Lower < QW < QW Upper	0.53	0.47
Yes	IncreaseCustomers	SU < SU Lower	QL > QL Upper	QW < QW Lower	0.50	0.50
Yes	IncreaseCustomers	SU < SU Lower	QL > QL Upper	QW > QW Upper	0.56	0.44
Yes	IncreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.30	0.70
Yes	IncreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.27	0.73
Yes	IncreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.33	0.67
Yes	IncreaseCustomers	SU > SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.27	0.73
Yes	IncreaseCustomers	SU > SU Upper	QL < QL Lower	QW < QW Lower	0.24	0.76
Yes	IncreaseCustomers	SU > SU Upper	QL < QL Lower	QW > QW Upper	0.30	0.70
Yes	IncreaseCustomers	SU > SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.33	0.67
Yes	IncreaseCustomers	SU > SU Upper	QL > QL Upper	QW < QW Lower	0.30	0.70
Yes	IncreaseCustomers	SU > SU Upper	QL > QL Upper	QW > QW Upper	0.36	0.64
Yes	DecreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.10	0.90
Yes	DecreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.07	0.93
Yes	DecreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.13	0.87

					IncreaseServers	
Improvement	LastAction	SU	QL	QW	Yes	No
Yes	DecreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.07	0.93
Yes	DecreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW < QW Lower	0.40	0.60
Yes	DecreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW > QW Upper	0.10	0.90
Yes	DecreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.13	0.87
Yes	DecreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW < QW Lower	0.10	0.90
Yes	DecreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW > QW Upper	0.16	0.84
Yes	DecreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.20	0.80
Yes	DecreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW < QW Lower	0.17	0.83
Yes	DecreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW > QW Upper	0.23	0.77
Yes	DecreaseServers	SU < SU Lower	QL < QL Lower	QW Lower < QW < QW Upper	0.17	0.83
Yes	DecreaseServers	SU < SU Lower	QL < QL Lower	QW < QW Lower	0.14	0.86
Yes	DecreaseServers	SU < SU Lower	QL < QL Lower	QW > QW Upper	0.20	0.80
Yes	DecreaseServers	SU < SU Lower	QL > QL Upper	QW Lower < QW < QW Upper	0.23	0.77
Yes	DecreaseServers	SU < SU Lower	QL > QL Upper	QW < QW Lower	0.20	0.80
Yes	DecreaseServers	SU < SU Lower	QL > QL Upper	QW > QW Upper	0.26	0.74
Yes	DecreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.05	0.95
Yes	DecreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.02	0.98
Yes	DecreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.08	0.92
Yes	DecreaseServers	SU > SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.02	0.98
Yes	DecreaseServers	SU > SU Upper	QL < QL Lower	QW < QW Lower	0.01	0.99
Yes	DecreaseServers	SU > SU Upper	QL < QL Lower	QW > QW Upper	0.05	0.95
Yes	DecreaseServers	SU > SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.08	0.92
Yes	DecreaseServers	SU > SU Upper	QL > QL Upper	QW < QW Lower	0.05	0.95
Yes	DecreaseServers	SU > SU Upper	QL > QL Upper	QW > QW Upper	0.11	0.89
Yes	IncreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.90	0.10
Yes	IncreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.87	0.13
Yes	IncreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.93	0.07
Yes	IncreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.87	0.13
Yes	IncreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW < QW Lower	0.84	0.16
Yes	IncreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW > QW Upper	0.90	0.10
Yes	IncreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.93	0.07
Yes	IncreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW < QW Lower	0.90	0.10
Yes	IncreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW > QW Upper	0.96	0.04

Improvement	LastAction	SU	QL	QW	IncreaseServers	
					Yes	No
Yes	IncreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.95	0.05
Yes	IncreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW < QW Lower	0.92	0.08
Yes	IncreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW > QW Upper	0.98	0.02
Yes	IncreaseServers	SU < SU Lower	QL < QL Lower	QW Lower < QW < QW Upper	0.92	0.08
Yes	IncreaseServers	SU < SU Lower	QL < QL Lower	QW < QW Lower	0.89	0.11
Yes	IncreaseServers	SU < SU Lower	QL < QL Lower	QW > QW Upper	0.95	0.05
Yes	IncreaseServers	SU < SU Lower	QL > QL Upper	QW Lower < QW < QW Upper	0.98	0.02
Yes	IncreaseServers	SU < SU Lower	QL > QL Upper	QW < QW Lower	0.95	0.05
Yes	IncreaseServers	SU < SU Lower	QL > QL Upper	QW > QW Upper	0.99	0.01
Yes	IncreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.80	0.20
Yes	IncreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.77	0.23
Yes	IncreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.83	0.17
Yes	IncreaseServers	SU > SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.75	0.25
Yes	IncreaseServers	SU > SU Upper	QL < QL Lower	QW < QW Lower	0.72	0.28
Yes	IncreaseServers	SU > SU Upper	QL < QL Lower	QW > QW Upper	0.78	0.22
Yes	IncreaseServers	SU > SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.85	0.15
Yes	IncreaseServers	SU > SU Upper	QL > QL Upper	QW < QW Lower	0.82	0.18
Yes	IncreaseServers	SU > SU Upper	QL > QL Upper	QW > QW Upper	0.88	0.12
No	DecreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.40	0.60
No	DecreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.37	0.63
No	DecreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.43	0.57
No	DecreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.35	0.65
No	DecreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW < QW Lower	0.32	0.68
No	DecreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW > QW Upper	0.38	0.62
No	DecreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.45	0.55
No	DecreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW < QW Lower	0.42	0.58
No	DecreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW > QW Upper	0.48	0.52
No	DecreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.50	0.50
No	DecreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW < QW Lower	0.47	0.53
No	DecreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW > QW Upper	0.53	0.47
No	DecreaseCustomers	SU < SU Lower	QL < QL Lower	QW Lower < QW < QW Upper	0.45	0.55
No	DecreaseCustomers	SU < SU Lower	QL < QL Lower	QW < QW Lower	0.42	0.58
No	DecreaseCustomers	SU < SU Lower	QL < QL Lower	QW > QW Upper	0.48	0.52

					IncreaseServers	
Improvement	LastAction	SU	QL	QW	Yes	No
No	DecreaseCustomers	SU < SU Lower	QL > QL Upper	QW Lower < QW < QW Upper	0.55	0.45
No	DecreaseCustomers	SU < SU Lower	QL > QL Upper	QW < QW Lower	0.52	0.48
No	DecreaseCustomers	SU < SU Lower	QL > QL Upper	QW > QW Upper	0.58	0.42
No	DecreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.30	0.70
No	DecreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.27	0.73
No	DecreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.33	0.67
No	DecreaseCustomers	SU > SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.25	0.75
No	DecreaseCustomers	SU > SU Upper	QL < QL Lower	QW < QW Lower	0.23	0.77
No	DecreaseCustomers	SU > SU Upper	QL < QL Lower	QW > QW Upper	0.28	0.72
No	DecreaseCustomers	SU > SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.35	0.65
No	DecreaseCustomers	SU > SU Upper	QL > QL Upper	QW < QW Lower	0.32	0.68
No	DecreaseCustomers	SU > SU Upper	QL > QL Upper	QW > QW Upper	0.38	0.62
No	IncreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.60	0.40
No	IncreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.57	0.43
No	IncreaseCustomers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.63	0.37
No	IncreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.65	0.35
No	IncreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW < QW Lower	0.62	0.38
No	IncreaseCustomers	SU Lower < SU < SU Upper	QL < QL Lower	QW > QW Upper	0.68	0.32
No	IncreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.55	0.45
No	IncreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW < QW Lower	0.52	0.48
No	IncreaseCustomers	SU Lower < SU < SU Upper	QL > QL Upper	QW > QW Upper	0.58	0.42
No	IncreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.70	0.30
No	IncreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW < QW Lower	0.67	0.33
No	IncreaseCustomers	SU < SU Lower	QL Lower < QL < QL Upper	QW > QW Upper	0.73	0.27
No	IncreaseCustomers	SU < SU Lower	QL < QL Lower	QW Lower < QW < QW Upper	0.75	0.25
No	IncreaseCustomers	SU < SU Lower	QL < QL Lower	QW < QW Lower	0.72	0.28
No	IncreaseCustomers	SU < SU Lower	QL < QL Lower	QW > QW Upper	0.78	0.22
No	IncreaseCustomers	SU < SU Lower	QL > QL Upper	QW Lower < QW < QW Upper	0.65	0.35
No	IncreaseCustomers	SU < SU Lower	QL > QL Upper	QW < QW Lower	0.62	0.38
No	IncreaseCustomers	SU < SU Lower	QL > QL Upper	QW > QW Upper	0.68	0.32
No	IncreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.50	0.50
No	IncreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.47	0.53
No	IncreaseCustomers	SU > SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.53	0.47
No	IncreaseCustomers	SU > SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.45	0.55

					IncreaseServers	
Improvement	LastAction	SU	QL	QW	Yes	No
No	IncreaseCustomers	SU > SU Upper	QL < QL Lower	QW < QW Lower	0.42	0.58
No	IncreaseCustomers	SU > SU Upper	QL < QL Lower	QW > QW Upper	0.48	0.52
No	IncreaseCustomers	SU > SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.55	0.45
No	IncreaseCustomers	SU > SU Upper	QL > QL Upper	QW < QW Lower	0.52	0.48
No	IncreaseCustomers	SU > SU Upper	QL > QL Upper	QW > QW Upper	0.58	0.42
No	DecreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.90	0.10
No	DecreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.87	0.13
No	DecreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.93	0.07
No	DecreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.85	0.15
No	DecreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW < QW Lower	0.82	0.18
No	DecreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW > QW Upper	0.88	0.12
No	DecreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.95	0.05
No	DecreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW < QW Lower	0.92	0.08
No	DecreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW > QW Upper	0.98	0.02
No	DecreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.95	0.05
No	DecreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW < QW Lower	0.92	0.08
No	DecreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW > QW Upper	0.98	0.02
No	DecreaseServers	SU < SU Lower	QL < QL Lower	QW Lower < QW < QW Upper	0.90	0.10
No	DecreaseServers	SU < SU Lower	QL < QL Lower	QW < QW Lower	0.87	0.13
No	DecreaseServers	SU < SU Lower	QL < QL Lower	QW > QW Upper	0.93	0.07
No	DecreaseServers	SU < SU Lower	QL > QL Upper	QW Lower < QW < QW Upper	0.98	0.02
No	DecreaseServers	SU < SU Lower	QL > QL Upper	QW < QW Lower	0.95	0.05
No	DecreaseServers	SU < SU Lower	QL > QL Upper	QW > QW Upper	0.99	0.01
No	DecreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.80	0.20
No	DecreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.77	0.23
No	DecreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.83	0.17
No	DecreaseServers	SU > SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.75	0.25
No	DecreaseServers	SU > SU Upper	QL < QL Lower	QW < QW Lower	0.72	0.28
No	DecreaseServers	SU > SU Upper	QL < QL Lower	QW > QW Upper	0.78	0.22
No	DecreaseServers	SU > SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.85	0.15
No	DecreaseServers	SU > SU Upper	QL > QL Upper	QW < QW Lower	0.82	0.18
No	DecreaseServers	SU > SU Upper	QL > QL Upper	QW > QW Upper	0.88	0.12
No	IncreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.10	0.90
No	IncreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.07	0.93

					IncreaseServers	
Improvement	LastAction	SU	QL	QW	Yes	No
No	IncreaseServers	SU Lower < SU < SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.13	0.87
No	IncreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.05	0.95
No	IncreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW < QW Lower	0.02	0.98
No	IncreaseServers	SU Lower < SU < SU Upper	QL < QL Lower	QW > QW Upper	0.08	0.92
No	IncreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.15	0.85
No	IncreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW < QW Lower	0.12	0.88
No	IncreaseServers	SU Lower < SU < SU Upper	QL > QL Upper	QW > QW Upper	0.18	0.82
No	IncreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.20	0.80
No	IncreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW < QW Lower	0.17	0.83
No	IncreaseServers	SU < SU Lower	QL Lower < QL < QL Upper	QW > QW Upper	0.23	0.77
No	IncreaseServers	SU < SU Lower	QL < QL Lower	QW Lower < QW < QW Upper	0.15	0.85
No	IncreaseServers	SU < SU Lower	QL < QL Lower	QW < QW Lower	0.12	0.88
No	IncreaseServers	SU < SU Lower	QL < QL Lower	QW > QW Upper	0.18	0.82
No	IncreaseServers	SU < SU Lower	QL > QL Upper	QW Lower < QW < QW Upper	0.25	0.75
No	IncreaseServers	SU < SU Lower	QL > QL Upper	QW < QW Lower	0.22	0.78
No	IncreaseServers	SU < SU Lower	QL > QL Upper	QW > QW Upper	0.28	0.72
No	IncreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW Lower < QW < QW Upper	0.05	0.95
No	IncreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW < QW Lower	0.02	0.98
No	IncreaseServers	SU > SU Upper	QL Lower < QL < QL Upper	QW > QW Upper	0.08	0.92
No	IncreaseServers	SU > SU Upper	QL < QL Lower	QW Lower < QW < QW Upper	0.03	0.97
No	IncreaseServers	SU > SU Upper	QL < QL Lower	QW < QW Lower	0.01	0.99
No	IncreaseServers	SU > SU Upper	QL < QL Lower	QW > QW Upper	0.05	0.95
No	IncreaseServers	SU > SU Upper	QL > QL Upper	QW Lower < QW < QW Upper	0.10	0.90
No	IncreaseServers	SU > SU Upper	QL > QL Upper	QW < QW Lower	0.07	0.93
No	IncreaseServers	SU > SU Upper	QL > QL Upper	QW > QW Upper	0.13	0.87

A3.2.3 TrainNest Belief Network

The "TrainNest" belief network was designed for use by the operations slave agent in the North Edmonton Sanitary Trunk (Nest) tunnel construction simulation presented in Chapter 4 and Appendix 8. The following information provides the necessary data to construct the belief network.

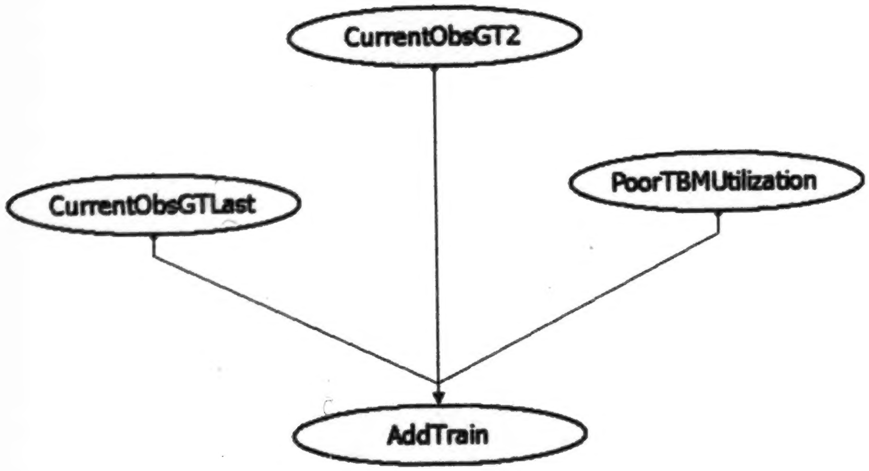

CurrentObsGT2
CurrentObsGTLast
PoorTBMUtilization
AddTrain

Figure A3-3. TrainNest Belief Network

Table A3-17. Belief Network Node Property Descriptions Table

Node	Property Description
CurrentObsGTLast	Input
CurrentObsGT2	Input
PoorTBMUtilization	Input
AddTrain	Output

Table A3-18. CurrentOBSGTLast Node Conditional Probability Table

CurrentObsGTLast	
Yes	0.5
No	0.5

Table A3-19. CurrentOBSGT2 Node Conditional Probability Table

CurrentObsGT2	
Yes	0.5
No	0.5

Table A3-20. PoorTBMUtlization Node Conditional Probability Table

PoorTBMUtilization	
Yes	0.5
No	0.5

Table A3-21. AddTrain Node Conditional Probability Table

			AddTrain	
CurrentObsGTLast	CurrentObsGT2	PoorTBMUtilization	Yes	No
Yes	Yes	Yes	1.0	0.0
Yes	Yes	No	0.6	0.4
Yes	No	Yes	0.6	0.4
Yes	No	No	0.3	0.7
No	Yes	Yes	0.6	0.4
No	Yes	No	0.3	0.7
No	No	Yes	0.3	0.7
No	No	No	0.0	1.0

A3.2.4 Lean Belief Network

The “lean” belief network was designed for use by the master agent in the North Edmonton Sanitary Trunk (Nest) tunnel construction simulation presented in Chapter 4 and Appendix 8. The following information provides the necessary data to construct the belief network.

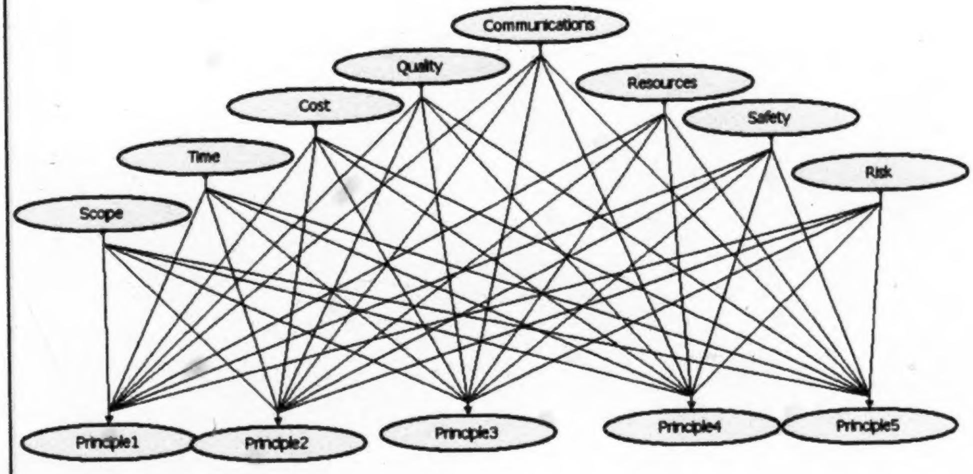

Communications
Quality
Resources
Cost
Safety
Time
Risk
Scope
Principle1
Principle2
Principle3
Principle4
Principle5

Figure A3-4. Lean Belief Network

Table A3-22. Belief Network Node Property Descriptions Table

Node	Property Description
Scope	Input
Time	Input
Cost	Input
Quality	Input
Communications	Input
Resources	Input
Safety	Input
Risk	Input
Principle1	Output
Principle2	Output
Principle3	Output
Principle4	Output
Principle5	Output

Table A3-23. Scope Node Conditional Probability Table

Scope	
Yes	0.5
No	0.5

Table A3-24. Time Node Conditional Probability Table

Time	
Yes	0.5
No	0.5

Table A3-25. Cost Node Conditional Probability Table

Cost	
Yes	0.5
No	0.5

Table A3-26. Quality Node Conditional Probability Table

Quality	
Yes	0.5
No	0.5

Table A3-27. Communications Node Conditional Probability Table

Communications	
Yes	0.5
No	0.5

Table A3-28. Resources Node Conditional Probability Table

Resources	
Yes	0.5
No	0.5

Table A3-29. Safety Node Conditional Probability Table

Safety	
Yes	0.5
No	0.5

Table A3-30. Risk Node Conditional Probability Table

Risk	
Yes	0.5
No	0.5

Table A3-31. Priciple1 Node Conditional Probability Table

								Principle 1	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	1	0
Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	1	0
Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	1	0
Yes	Yes	Yes	Yes	Yes	Yes	No	No	1	0
Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	1	0
Yes	Yes	Yes	Yes	Yes	No	Yes	No	1	0
Yes	Yes	Yes	Yes	Yes	No	No	Yes	1	0
Yes	Yes	Yes	Yes	Yes	No	No	No	1	0
Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	1	0
Yes	Yes	Yes	Yes	No	Yes	Yes	No	1	0
Yes	Yes	Yes	Yes	No	Yes	No	Yes	1	0
Yes	Yes	Yes	Yes	No	Yes	No	No	1	0
Yes	Yes	Yes	Yes	No	No	Yes	Yes	1	0
Yes	Yes	Yes	Yes	No	No	Yes	No	1	0
Yes	Yes	Yes	Yes	No	No	No	Yes	1	0
Yes	Yes	Yes	Yes	No	No	No	No	1	0
Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	0	1
Yes	Yes	Yes	No	Yes	Yes	Yes	No	0	1
Yes	Yes	Yes	No	Yes	Yes	No	Yes	0	1
Yes	Yes	Yes	No	Yes	Yes	No	No	0	1
Yes	Yes	Yes	No	Yes	No	Yes	Yes	0	1
Yes	Yes	Yes	No	Yes	No	Yes	No	0	1
Yes	Yes	Yes	No	Yes	No	No	Yes	0	1
Yes	Yes	Yes	No	Yes	No	No	No	0	1
Yes	Yes	Yes	No	No	Yes	Yes	Yes	0	1
Yes	Yes	Yes	No	No	Yes	Yes	No	0	1
Yes	Yes	Yes	No	No	Yes	No	Yes	0	1
Yes	Yes	Yes	No	No	Yes	No	No	0	1
Yes	Yes	Yes	No	No	No	Yes	Yes	0	1
Yes	Yes	Yes	No	No	No	Yes	No	0	1
Yes	Yes	Yes	No	No	No	No	Yes	0	1
Yes	Yes	Yes	No	No	No	No	No	0	1
Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	1	0
Yes	Yes	No	Yes	Yes	Yes	Yes	No	1	0
Yes	Yes	No	Yes	Yes	Yes	No	Yes	1	0
Yes	Yes	No	Yes	Yes	Yes	No	No	1	0

								Principle 1	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
Yes	Yes	No	Yes	Yes	No	Yes	Yes	1	0
Yes	Yes	No	Yes	Yes	No	Yes	No	1	0
Yes	Yes	No	Yes	Yes	No	No	Yes	1	0
Yes	Yes	No	Yes	Yes	No	No	No	1	0
Yes	Yes	No	Yes	No	Yes	Yes	Yes	1	0
Yes	Yes	No	Yes	No	Yes	Yes	No	1	0
Yes	Yes	No	Yes	No	Yes	No	Yes	1	0
Yes	Yes	No	Yes	No	Yes	No	No	1	0
Yes	Yes	No	Yes	No	No	Yes	Yes	1	0
Yes	Yes	No	Yes	No	No	Yes	No	1	0
Yes	Yes	No	Yes	No	No	No	Yes	1	0
Yes	Yes	No	Yes	No	No	No	No	1	0
Yes	Yes	No	No	Yes	Yes	Yes	Yes	0	1
Yes	Yes	No	No	Yes	Yes	Yes	No	0	1
Yes	Yes	No	No	Yes	Yes	No	Yes	0	1
Yes	Yes	No	No	Yes	Yes	No	No	0	1
Yes	Yes	No	No	Yes	No	Yes	Yes	0	1
Yes	Yes	No	No	Yes	No	Yes	No	0	1
Yes	Yes	No	No	Yes	No	No	Yes	0	1
Yes	Yes	No	No	Yes	No	No	No	0	1
Yes	Yes	No	No	No	Yes	Yes	Yes	0	1
Yes	Yes	No	No	No	Yes	Yes	No	0	1
Yes	Yes	No	No	No	Yes	No	Yes	0	1
Yes	Yes	No	No	No	Yes	No	No	0	1
Yes	Yes	No	No	No	No	Yes	Yes	0	1
Yes	Yes	No	No	No	No	Yes	No	0	1
Yes	Yes	No	No	No	No	No	Yes	0	1
Yes	Yes	No	No	No	No	No	No	0	1
Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	1	0
Yes	No	Yes	Yes	Yes	Yes	Yes	No	1	0
Yes	No	Yes	Yes	Yes	Yes	No	Yes	1	0
Yes	No	Yes	Yes	Yes	Yes	No	No	1	0
Yes	No	Yes	Yes	Yes	No	Yes	Yes	1	0
Yes	No	Yes	Yes	Yes	No	Yes	No	1	0
Yes	No	Yes	Yes	Yes	No	No	Yes	1	0
Yes	No	Yes	Yes	Yes	No	No	No	1	0
Yes	No	Yes	Yes	No	Yes	Yes	Yes	1	0
Yes	No	Yes	Yes	No	Yes	Yes	No	1	0
Yes	No	Yes	Yes	No	Yes	No	Yes	1	0
Yes	No	Yes	Yes	No	Yes	No	No	1	0
Yes	No	Yes	Yes	No	No	Yes	Yes	1	0
Yes	No	Yes	Yes	No	No	Yes	No	1	0
Yes	No	Yes	Yes	No	No	No	Yes	1	0
Yes	No	Yes	Yes	No	No	No	No	1	0
Yes	No	Yes	No	Yes	Yes	Yes	Yes	0	1
Yes	No	Yes	No	Yes	Yes	Yes	No	0	1

								Principle 1	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
Yes	No	Yes	No	Yes	Yes	No	Yes	0	1
Yes	No	Yes	No	Yes	Yes	No	No	0	1
Yes	No	Yes	No	Yes	No	Yes	Yes	0	1
Yes	No	Yes	No	Yes	No	Yes	No	0	1
Yes	No	Yes	No	Yes	No	No	Yes	0	1
Yes	No	Yes	No	Yes	No	No	No	0	1
Yes	No	Yes	No	No	Yes	Yes	Yes	0	1
Yes	No	Yes	No	No	Yes	Yes	No	0	1
Yes	No	Yes	No	No	Yes	No	Yes	0	1
Yes	No	Yes	No	No	Yes	No	No	0	1
Yes	No	Yes	No	No	No	Yes	Yes	0	1
Yes	No	Yes	No	No	No	Yes	No	0	1
Yes	No	Yes	No	No	No	No	Yes	0	1
Yes	No	Yes	No	No	No	No	No	0	1
Yes	No	No	Yes	Yes	Yes	Yes	Yes	1	0
Yes	No	No	Yes	Yes	Yes	Yes	No	1	0
Yes	No	No	Yes	Yes	Yes	No	Yes	1	0
Yes	No	No	Yes	Yes	Yes	No	No	1	0
Yes	No	No	Yes	Yes	No	Yes	Yes	1	0
Yes	No	No	Yes	Yes	No	Yes	No	1	0
Yes	No	No	Yes	Yes	No	No	Yes	1	0
Yes	No	No	Yes	Yes	No	No	No	1	0
Yes	No	No	Yes	No	Yes	Yes	Yes	1	0
Yes	No	No	Yes	No	Yes	Yes	No	1	0
Yes	No	No	Yes	No	Yes	No	Yes	1	0
Yes	No	No	Yes	No	Yes	No	No	1	0
Yes	No	No	Yes	No	No	Yes	Yes	1	0
Yes	No	No	Yes	No	No	Yes	No	1	0
Yes	No	No	Yes	No	No	No	Yes	1	0
Yes	No	No	Yes	No	No	No	No	1	0
Yes	No	No	No	Yes	Yes	Yes	Yes	1	0
Yes	No	No	No	Yes	Yes	Yes	No	1	0
Yes	No	No	No	Yes	Yes	No	Yes	1	0
Yes	No	No	No	Yes	Yes	No	No	1	0
Yes	No	No	No	Yes	No	Yes	Yes	1	0
Yes	No	No	No	Yes	No	Yes	No	1	0
Yes	No	No	No	Yes	No	No	Yes	1	0
Yes	No	No	No	Yes	No	No	No	1	0
Yes	No	No	No	No	Yes	Yes	Yes	1	0
Yes	No	No	No	No	Yes	Yes	No	1	0
Yes	No	No	No	No	Yes	No	Yes	1	0
Yes	No	No	No	No	Yes	No	No	1	0
Yes	No	No	No	No	No	Yes	Yes	1	0
Yes	No	No	No	No	No	Yes	No	1	0
Yes	No	No	No	No	No	No	Yes	1	0
Yes	No	No	No	No	No	No	No	0	1

								Principle 1	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	1	0
No	Yes	Yes	Yes	Yes	Yes	Yes	No	1	0
No	Yes	Yes	Yes	Yes	Yes	No	Yes	1	0
No	Yes	Yes	Yes	Yes	Yes	No	No	1	0
No	Yes	Yes	Yes	Yes	No	Yes	Yes	1	0
No	Yes	Yes	Yes	Yes	No	Yes	No	1	0
No	Yes	Yes	Yes	Yes	No	No	Yes	1	0
No	Yes	Yes	Yes	Yes	No	No	No	1	0
No	Yes	Yes	Yes	No	Yes	Yes	Yes	1	0
No	Yes	Yes	Yes	No	Yes	Yes	No	1	0
No	Yes	Yes	Yes	No	Yes	No	Yes	1	0
No	Yes	Yes	Yes	No	Yes	No	No	1	0
No	Yes	Yes	Yes	No	No	Yes	Yes	1	0
No	Yes	Yes	Yes	No	No	Yes	No	1	0
No	Yes	Yes	Yes	No	No	No	Yes	1	0
No	Yes	Yes	Yes	No	No	No	No	1	0
No	Yes	Yes	No	Yes	Yes	Yes	Yes	0	1
No	Yes	Yes	No	Yes	Yes	Yes	No	0	1
No	Yes	Yes	No	Yes	Yes	No	Yes	0	1
No	Yes	Yes	No	Yes	Yes	No	No	0	1
No	Yes	Yes	No	Yes	No	Yes	Yes	0	1
No	Yes	Yes	No	Yes	No	Yes	No	0	1
No	Yes	Yes	No	Yes	No	No	Yes	0	1
No	Yes	Yes	No	Yes	No	No	No	0	1
No	Yes	Yes	No	No	Yes	Yes	Yes	0	1
No	Yes	Yes	No	No	Yes	Yes	No	0	1
No	Yes	Yes	No	No	Yes	No	Yes	0	1
No	Yes	Yes	No	No	Yes	No	No	0	1
No	Yes	Yes	No	No	No	Yes	Yes	0	1
No	Yes	Yes	No	No	No	Yes	No	0	1
No	Yes	Yes	No	No	No	No	Yes	0	1
No	Yes	Yes	No	No	No	No	No	0	1
No	Yes	No	Yes	Yes	Yes	Yes	Yes	1	0
No	Yes	No	Yes	Yes	Yes	Yes	No	1	0
No	Yes	No	Yes	Yes	Yes	No	Yes	1	0
No	Yes	No	Yes	Yes	Yes	No	No	1	0
No	Yes	No	Yes	Yes	No	Yes	Yes	1	0
No	Yes	No	Yes	Yes	No	Yes	No	1	0
No	Yes	No	Yes	Yes	No	No	Yes	1	0
No	Yes	No	Yes	Yes	No	No	No	1	0
No	Yes	No	Yes	No	Yes	Yes	Yes	1	0
No	Yes	No	Yes	No	Yes	Yes	No	1	0
No	Yes	No	Yes	No	Yes	No	Yes	1	0
No	Yes	No	Yes	No	Yes	No	No	1	0
No	Yes	No	Yes	No	No	Yes	Yes	1	0
No	Yes	No	Yes	No	No	Yes	No	1	0

								Principle 1	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
No	Yes	No	Yes	No	No	No	Yes	1	0
No	Yes	No	Yes	No	No	No	No	1	0
No	Yes	No	No	Yes	Yes	Yes	Yes	0	1
No	Yes	No	No	Yes	Yes	Yes	No	0	1
No	Yes	No	No	Yes	Yes	No	Yes	0	1
No	Yes	No	No	Yes	Yes	No	No	0	1
No	Yes	No	No	Yes	No	Yes	Yes	0	1
No	Yes	No	No	Yes	No	Yes	No	0	1
No	Yes	No	No	Yes	No	No	Yes	0	1
No	Yes	No	No	Yes	No	No	No	0	1
No	Yes	No	No	No	Yes	Yes	Yes	0	1
No	Yes	No	No	No	Yes	Yes	No	0	1
No	Yes	No	No	No	Yes	No	Yes	0	1
No	Yes	No	No	No	Yes	No	No	0	1
No	Yes	No	No	No	No	Yes	Yes	0	1
No	Yes	No	No	No	No	Yes	No	0	1
No	Yes	No	No	No	No	No	Yes	0	1
No	Yes	No	No	No	No	No	No	0	1
No	No	Yes	Yes	Yes	Yes	Yes	Yes	1	0
No	No	Yes	Yes	Yes	Yes	Yes	No	1	0
No	No	Yes	Yes	Yes	Yes	No	Yes	1	0
No	No	Yes	Yes	Yes	Yes	No	No	1	0
No	No	Yes	Yes	Yes	No	Yes	Yes	1	0
No	No	Yes	Yes	Yes	No	Yes	No	1	0
No	No	Yes	Yes	Yes	No	No	Yes	1	0
No	No	Yes	Yes	Yes	No	No	No	1	0
No	No	Yes	Yes	No	Yes	Yes	Yes	1	0
No	No	Yes	Yes	No	Yes	Yes	No	1	0
No	No	Yes	Yes	No	Yes	No	Yes	1	0
No	No	Yes	Yes	No	Yes	No	No	1	0
No	No	Yes	Yes	No	No	Yes	Yes	1	0
No	No	Yes	Yes	No	No	Yes	No	1	0
No	No	Yes	Yes	No	No	No	Yes	1	0
No	No	Yes	Yes	No	No	No	No	1	0
No	No	Yes	No	Yes	Yes	Yes	Yes	0	1
No	No	Yes	No	Yes	Yes	Yes	No	0	1
No	No	Yes	No	Yes	Yes	No	Yes	0	1
No	No	Yes	No	Yes	Yes	No	No	0	1
No	No	Yes	No	Yes	No	Yes	Yes	0	1
No	No	Yes	No	Yes	No	Yes	No	0	1
No	No	Yes	No	Yes	No	No	Yes	0	1
No	No	Yes	No	Yes	No	No	No	0	1
No	No	Yes	No	No	Yes	Yes	Yes	0	1
No	No	Yes	No	No	Yes	Yes	No	0	1
No	No	Yes	No	No	Yes	No	Yes	0	1
No	No	Yes	No	No	Yes	No	No	0	1

								Principle 1	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
No	No	Yes	No	No	No	Yes	Yes	0	1
No	No	Yes	No	No	No	Yes	No	0	1
No	No	Yes	No	No	No	No	Yes	0	1
No	No	Yes	No	No	No	No	No	0	1
No	No	No	Yes	Yes	Yes	Yes	Yes	1	0
No	No	No	Yes	Yes	Yes	Yes	No	1	0
No	No	No	Yes	Yes	Yes	No	Yes	1	0
No	No	No	Yes	Yes	Yes	No	No	1	0
No	No	No	Yes	Yes	No	Yes	Yes	1	0
No	No	No	Yes	Yes	No	Yes	No	1	0
No	No	No	Yes	Yes	No	No	Yes	1	0
No	No	No	Yes	Yes	No	No	No	1	0
No	No	No	Yes	No	Yes	Yes	Yes	1	0
No	No	No	Yes	No	Yes	Yes	No	1	0
No	No	No	Yes	No	Yes	No	Yes	1	0
No	No	No	Yes	No	Yes	No	No	1	0
No	No	No	Yes	No	No	Yes	Yes	1	0
No	No	No	Yes	No	No	Yes	No	1	0
No	No	No	Yes	No	No	No	Yes	1	0
No	No	No	Yes	No	No	No	No	0	1
No	No	No	No	Yes	Yes	Yes	Yes	0	1
No	No	No	No	Yes	Yes	Yes	No	0	1
No	No	No	No	Yes	Yes	No	Yes	0	1
No	No	No	No	Yes	Yes	No	No	0	1
No	No	No	No	Yes	No	Yes	Yes	0	1
No	No	No	No	Yes	No	Yes	No	0	1
No	No	No	No	Yes	No	No	Yes	0	1
No	No	No	No	Yes	No	No	No	0	1
No	No	No	No	No	Yes	Yes	Yes	0	1
No	No	No	No	No	Yes	Yes	No	0	1
No	No	No	No	No	Yes	No	Yes	0	1
No	No	No	No	No	Yes	No	No	0	1
No	No	No	No	No	No	Yes	Yes	0	1
No	No	No	No	No	No	Yes	No	0	1
No	No	No	No	No	No	No	Yes	0	1
No	No	No	No	No	No	No	No	0	1

Table A3-32. Priciple2 Node Conditional Probability Table

								Principle 2	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	1	0
Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	1	0
Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	1	0
Yes	Yes	Yes	Yes	Yes	Yes	No	No	1	0

								Principle 2	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	1	0
Yes	Yes	Yes	Yes	Yes	No	Yes	No	1	0
Yes	Yes	Yes	Yes	Yes	No	No	Yes	1	0
Yes	Yes	Yes	Yes	Yes	No	No	No	1	0
Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	1	0
Yes	Yes	Yes	Yes	No	Yes	Yes	No	1	0
Yes	Yes	Yes	Yes	No	Yes	No	Yes	1	0
Yes	Yes	Yes	Yes	No	Yes	No	No	1	0
Yes	Yes	Yes	Yes	No	No	Yes	Yes	1	0
Yes	Yes	Yes	Yes	No	No	Yes	No	1	0
Yes	Yes	Yes	Yes	No	No	No	Yes	1	0
Yes	Yes	Yes	Yes	No	No	No	No	1	0
Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	1	0
Yes	Yes	Yes	No	Yes	Yes	Yes	No	1	0
Yes	Yes	Yes	No	Yes	Yes	No	Yes	1	0
Yes	Yes	Yes	No	Yes	Yes	No	No	1	0
Yes	Yes	Yes	No	Yes	No	Yes	Yes	1	0
Yes	Yes	Yes	No	Yes	No	Yes	No	1	0
Yes	Yes	Yes	No	Yes	No	No	Yes	1	0
Yes	Yes	Yes	No	Yes	No	No	No	1	0
Yes	Yes	Yes	No	No	Yes	Yes	Yes	1	0
Yes	Yes	Yes	No	No	Yes	Yes	No	1	0
Yes	Yes	Yes	No	No	Yes	No	Yes	1	0
Yes	Yes	Yes	No	No	Yes	No	No	1	0
Yes	Yes	Yes	No	No	No	Yes	Yes	1	0
Yes	Yes	Yes	No	No	No	Yes	No	1	0
Yes	Yes	Yes	No	No	No	No	Yes	1	0
Yes	Yes	Yes	No	No	No	No	No	1	0
Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	1	0
Yes	Yes	No	Yes	Yes	Yes	Yes	No	1	0
Yes	Yes	No	Yes	Yes	Yes	No	Yes	1	0
Yes	Yes	No	Yes	Yes	Yes	No	No	1	0
Yes	Yes	No	Yes	Yes	No	Yes	Yes	1	0
Yes	Yes	No	Yes	Yes	No	Yes	No	1	0
Yes	Yes	No	Yes	Yes	No	No	Yes	1	0
Yes	Yes	No	Yes	Yes	No	No	No	1	0
Yes	Yes	No	Yes	No	Yes	Yes	Yes	1	0
Yes	Yes	No	Yes	No	Yes	Yes	No	1	0
Yes	Yes	No	Yes	No	Yes	No	Yes	1	0
Yes	Yes	No	Yes	No	Yes	No	No	1	0
Yes	Yes	No	Yes	No	No	Yes	Yes	1	0
Yes	Yes	No	Yes	No	No	Yes	No	1	0
Yes	Yes	No	Yes	No	No	No	Yes	1	0
Yes	Yes	No	Yes	No	No	No	No	1	0
Yes	Yes	No	No	Yes	Yes	Yes	Yes	1	0
Yes	Yes	No	No	Yes	Yes	Yes	No	1	0

Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Principle 2	
								Yes	No
Yes	Yes	No	No	Yes	Yes	No	Yes	1	0
Yes	Yes	No	No	Yes	Yes	No	No	1	0
Yes	Yes	No	No	Yes	No	Yes	Yes	1	0
Yes	Yes	No	No	Yes	No	Yes	No	1	0
Yes	Yes	No	No	Yes	No	No	Yes	1	0
Yes	Yes	No	No	Yes	No	No	No	1	0
Yes	Yes	No	No	No	Yes	Yes	Yes	1	0
Yes	Yes	No	No	No	Yes	Yes	No	1	0
Yes	Yes	No	No	No	Yes	No	Yes	1	0
Yes	Yes	No	No	No	Yes	No	No	1	0
Yes	Yes	No	No	No	No	Yes	Yes	1	0
Yes	Yes	No	No	No	No	Yes	No	1	0
Yes	Yes	No	No	No	No	No	Yes	1	0
Yes	Yes	No	No	No	No	No	No	1	0
Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	0	1
Yes	No	Yes	Yes	Yes	Yes	Yes	No	0	1
Yes	No	Yes	Yes	Yes	Yes	No	Yes	0	1
Yes	No	Yes	Yes	Yes	Yes	No	No	0	1
Yes	No	Yes	Yes	Yes	No	Yes	Yes	0	1
Yes	No	Yes	Yes	Yes	No	Yes	No	0	1
Yes	No	Yes	Yes	Yes	No	No	Yes	0	1
Yes	No	Yes	Yes	Yes	No	No	No	0	1
Yes	No	Yes	Yes	No	Yes	Yes	Yes	0	1
Yes	No	Yes	Yes	No	Yes	Yes	No	0	1
Yes	No	Yes	Yes	No	Yes	No	Yes	0	1
Yes	No	Yes	Yes	No	Yes	No	No	0	1
Yes	No	Yes	Yes	No	No	Yes	Yes	0	1
Yes	No	Yes	Yes	No	No	Yes	No	0	1
Yes	No	Yes	Yes	No	No	No	Yes	0	1
Yes	No	Yes	Yes	No	No	No	No	0	1
Yes	No	Yes	No	Yes	Yes	Yes	Yes	0	1
Yes	No	Yes	No	Yes	Yes	Yes	No	0	1
Yes	No	Yes	No	Yes	Yes	No	Yes	0	1
Yes	No	Yes	No	Yes	Yes	No	No	0	1
Yes	No	Yes	No	Yes	No	Yes	Yes	0	1
Yes	No	Yes	No	Yes	No	Yes	No	0	1
Yes	No	Yes	No	Yes	No	No	Yes	0	1
Yes	No	Yes	No	Yes	No	No	No	0	1
Yes	No	Yes	No	No	Yes	Yes	Yes	0	1
Yes	No	Yes	No	No	Yes	Yes	No	0	1
Yes	No	Yes	No	No	Yes	No	Yes	0	1
Yes	No	Yes	No	No	Yes	No	No	0	1
Yes	No	Yes	No	No	No	Yes	Yes	0	1
Yes	No	Yes	No	No	No	Yes	No	0	1
Yes	No	Yes	No	No	No	No	Yes	0	1
Yes	No	Yes	No	No	No	No	No	0	1

								Principle 2	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
Yes	No	No	Yes	Yes	Yes	Yes	Yes	0	1
Yes	No	No	Yes	Yes	Yes	Yes	No	0	1
Yes	No	No	Yes	Yes	Yes	No	Yes	0	1
Yes	No	No	Yes	Yes	Yes	No	No	0	1
Yes	No	No	Yes	Yes	No	Yes	Yes	0	1
Yes	No	No	Yes	Yes	No	Yes	No	0	1
Yes	No	No	Yes	Yes	No	No	Yes	0	1
Yes	No	No	Yes	Yes	No	No	No	0	1
Yes	No	No	Yes	No	Yes	Yes	Yes	0	1
Yes	No	No	Yes	No	Yes	Yes	No	0	1
Yes	No	No	Yes	No	Yes	No	Yes	0	1
Yes	No	No	Yes	No	Yes	No	No	0	1
Yes	No	No	Yes	No	No	Yes	Yes	0	1
Yes	No	No	Yes	No	No	Yes	No	0	1
Yes	No	No	Yes	No	No	No	Yes	0	1
Yes	No	No	Yes	No	No	No	No	0	1
Yes	No	No	No	Yes	Yes	Yes	Yes	0	1
Yes	No	No	No	Yes	Yes	Yes	No	0	1
Yes	No	No	No	Yes	Yes	No	Yes	0	1
Yes	No	No	No	Yes	Yes	No	No	0	1
Yes	No	No	No	Yes	No	Yes	Yes	0	1
Yes	No	No	No	Yes	No	Yes	No	0	1
Yes	No	No	No	Yes	No	No	Yes	0	1
Yes	No	No	No	Yes	No	No	No	0	1
Yes	No	No	No	No	Yes	Yes	Yes	0	1
Yes	No	No	No	No	Yes	Yes	No	0	1
Yes	No	No	No	No	Yes	No	Yes	0	1
Yes	No	No	No	No	Yes	No	No	0	1
Yes	No	No	No	No	No	Yes	Yes	0	1
Yes	No	No	No	No	No	Yes	No	0	1
Yes	No	No	No	No	No	No	Yes	0	1
Yes	No	No	No	No	No	No	No	0	1
No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	1	0
No	Yes	Yes	Yes	Yes	Yes	Yes	No	1	0
No	Yes	Yes	Yes	Yes	Yes	No	Yes	1	0
No	Yes	Yes	Yes	Yes	Yes	No	No	1	0
No	Yes	Yes	Yes	Yes	No	Yes	Yes	1	0
No	Yes	Yes	Yes	Yes	No	Yes	No	1	0
No	Yes	Yes	Yes	Yes	No	No	Yes	1	0
No	Yes	Yes	Yes	Yes	No	No	No	1	0
No	Yes	Yes	Yes	No	Yes	Yes	Yes	1	0
No	Yes	Yes	Yes	No	Yes	Yes	No	1	0
No	Yes	Yes	Yes	No	Yes	No	Yes	1	0
No	Yes	Yes	Yes	No	Yes	No	No	1	0
No	Yes	Yes	Yes	No	No	Yes	Yes	1	0
No	Yes	Yes	Yes	No	No	Yes	No	1	0

								Principle 2	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
No	Yes	Yes	Yes	No	No	No	Yes	1	0
No	Yes	Yes	Yes	No	No	No	No	1	0
No	Yes	Yes	No	Yes	Yes	Yes	Yes	1	0
No	Yes	Yes	No	Yes	Yes	Yes	No	1	0
No	Yes	Yes	No	Yes	Yes	No	Yes	1	0
No	Yes	Yes	No	Yes	Yes	No	No	1	0
No	Yes	Yes	No	Yes	No	Yes	Yes	1	0
No	Yes	Yes	No	Yes	No	Yes	No	1	0
No	Yes	Yes	No	Yes	No	No	Yes	1	0
No	Yes	Yes	No	Yes	No	No	No	1	0
No	Yes	Yes	No	No	Yes	Yes	Yes	1	0
No	Yes	Yes	No	No	Yes	Yes	No	1	0
No	Yes	Yes	No	No	Yes	No	Yes	1	0
No	Yes	Yes	No	No	Yes	No	No	1	0
No	Yes	Yes	No	No	No	Yes	Yes	1	0
No	Yes	Yes	No	No	No	Yes	No	1	0
No	Yes	Yes	No	No	No	No	Yes	1	0
No	Yes	Yes	No	No	No	No	No	1	0
No	Yes	No	Yes	Yes	Yes	Yes	Yes	1	0
No	Yes	No	Yes	Yes	Yes	Yes	No	1	0
No	Yes	No	Yes	Yes	Yes	No	Yes	1	0
No	Yes	No	Yes	Yes	Yes	No	No	1	0
No	Yes	No	Yes	Yes	No	Yes	Yes	1	0
No	Yes	No	Yes	Yes	No	Yes	No	1	0
No	Yes	No	Yes	Yes	No	No	Yes	1	0
No	Yes	No	Yes	Yes	No	No	No	1	0
No	Yes	No	Yes	No	Yes	Yes	Yes	1	0
No	Yes	No	Yes	No	Yes	Yes	No	1	0
No	Yes	No	Yes	No	Yes	No	Yes	1	0
No	Yes	No	Yes	No	Yes	No	No	1	0
No	Yes	No	Yes	No	No	Yes	Yes	1	0
No	Yes	No	Yes	No	No	Yes	No	1	0
No	Yes	No	Yes	No	No	No	Yes	1	0
No	Yes	No	Yes	No	No	No	No	1	0
No	Yes	No	No	Yes	Yes	Yes	Yes	1	0
No	Yes	No	No	Yes	Yes	Yes	No	1	0
No	Yes	No	No	Yes	Yes	No	Yes	1	0
No	Yes	No	No	Yes	Yes	No	No	1	0
No	Yes	No	No	Yes	No	Yes	Yes	1	0
No	Yes	No	No	Yes	No	Yes	No	1	0
No	Yes	No	No	Yes	No	No	Yes	1	0
No	Yes	No	No	Yes	No	No	No	1	0
No	Yes	No	No	No	Yes	Yes	Yes	1	0
No	Yes	No	No	No	Yes	Yes	No	1	0
No	Yes	No	No	No	Yes	No	Yes	1	0
No	Yes	No	No	No	Yes	No	No	1	0

								Principle 2	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
No	Yes	No	No	No	No	Yes	Yes	1	0
No	Yes	No	No	No	No	Yes	No	1	0
No	Yes	No	No	No	No	No	Yes	1	0
No	Yes	No	No	No	No	No	No	0	1
No	No	Yes	Yes	Yes	Yes	Yes	Yes	0	1
No	No	Yes	Yes	Yes	Yes	Yes	No	0	1
No	No	Yes	Yes	Yes	Yes	No	Yes	0	1
No	No	Yes	Yes	Yes	Yes	No	No	0	1
No	No	Yes	Yes	Yes	No	Yes	Yes	0	1
No	No	Yes	Yes	Yes	No	Yes	No	0	1
No	No	Yes	Yes	Yes	No	No	Yes	0	1
No	No	Yes	Yes	Yes	No	No	No	0	1
No	No	Yes	Yes	No	Yes	Yes	Yes	0	1
No	No	Yes	Yes	No	Yes	Yes	No	0	1
No	No	Yes	Yes	No	Yes	No	Yes	0	1
No	No	Yes	Yes	No	Yes	No	No	0	1
No	No	Yes	Yes	No	No	Yes	Yes	0	1
No	No	Yes	Yes	No	No	Yes	No	0	1
No	No	Yes	Yes	No	No	No	Yes	0	1
No	No	Yes	Yes	No	No	No	No	0	1
No	No	Yes	No	Yes	Yes	Yes	Yes	0	1
No	No	Yes	No	Yes	Yes	Yes	No	0	1
No	No	Yes	No	Yes	Yes	No	Yes	0	1
No	No	Yes	No	Yes	Yes	No	No	0	1
No	No	Yes	No	Yes	No	Yes	Yes	0	1
No	No	Yes	No	Yes	No	Yes	No	0	1
No	No	Yes	No	Yes	No	No	Yes	0	1
No	No	Yes	No	Yes	No	No	No	0	1
No	No	Yes	No	No	Yes	Yes	Yes	0	1
No	No	Yes	No	No	Yes	Yes	No	0	1
No	No	Yes	No	No	Yes	No	Yes	0	1
No	No	Yes	No	No	Yes	No	No	0	1
No	No	Yes	No	No	No	Yes	Yes	0	1
No	No	Yes	No	No	No	Yes	No	0	1
No	No	Yes	No	No	No	No	Yes	0	1
No	No	Yes	No	No	No	No	No	0	1
No	No	No	Yes	Yes	Yes	Yes	Yes	0	1
No	No	No	Yes	Yes	Yes	Yes	No	0	1
No	No	No	Yes	Yes	Yes	No	Yes	0	1
No	No	No	Yes	Yes	Yes	No	No	0	1
No	No	No	Yes	Yes	No	Yes	Yes	0	1
No	No	No	Yes	Yes	No	Yes	No	0	1
No	No	No	Yes	Yes	No	No	Yes	0	1
No	No	No	Yes	Yes	No	No	No	0	1
No	No	No	Yes	No	Yes	Yes	Yes	0	1
No	No	No	Yes	No	Yes	Yes	No	0	1

								Principle 2	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
No	No	No	Yes	No	Yes	No	Yes	0	1
No	No	No	Yes	No	Yes	No	No	0	1
No	No	No	Yes	No	No	Yes	Yes	0	1
No	No	No	Yes	No	No	Yes	No	0	1
No	No	No	Yes	No	No	No	Yes	0	1
No	No	No	Yes	No	No	No	No	0	1
No	No	No	No	Yes	Yes	Yes	Yes	0	1
No	No	No	No	Yes	Yes	Yes	No	0	1
No	No	No	No	Yes	Yes	No	Yes	0	1
No	No	No	No	Yes	Yes	No	No	0	1
No	No	No	No	Yes	No	Yes	Yes	0	1
No	No	No	No	Yes	No	Yes	No	0	1
No	No	No	No	Yes	No	No	Yes	0	1
No	No	No	No	Yes	No	No	No	0	1
No	No	No	No	No	Yes	Yes	Yes	0	1
No	No	No	No	No	Yes	Yes	No	0	1
No	No	No	No	No	Yes	No	Yes	0	1
No	No	No	No	No	Yes	No	No	0	1
No	No	No	No	No	No	Yes	Yes	0	1
No	No	No	No	No	No	Yes	No	0	1
No	No	No	No	No	No	No	Yes	0	1
No	No	No	No	No	No	No	No	0	1

Table A3-33. Priciple3 Node Conditional Probability Table

								Principle 3	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	0.95	0.05
Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	0.95	0.05
Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	0.95	0.05
Yes	Yes	Yes	Yes	Yes	Yes	No	No	0.95	0.05
Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	0.95	0.05
Yes	Yes	Yes	Yes	Yes	No	Yes	No	0.95	0.05
Yes	Yes	Yes	Yes	Yes	No	No	Yes	0.95	0.05
Yes	Yes	Yes	Yes	Yes	No	No	No	0.95	0.05
Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	0.95	0.05
Yes	Yes	Yes	Yes	No	Yes	Yes	No	0.95	0.05
Yes	Yes	Yes	Yes	No	Yes	No	Yes	0.95	0.05
Yes	Yes	Yes	Yes	No	Yes	No	No	0.95	0.05
Yes	Yes	Yes	Yes	No	No	Yes	Yes	0.95	0.05
Yes	Yes	Yes	Yes	No	No	Yes	No	0.95	0.05
Yes	Yes	Yes	Yes	No	No	No	Yes	0.95	0.05
Yes	Yes	Yes	Yes	No	No	No	No	0.95	0.05
Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	0.95	0.05
Yes	Yes	Yes	No	Yes	Yes	Yes	No	0.95	0.05

								Principle 3	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
Yes	Yes	Yes	No	Yes	Yes	No	Yes	0.95	0.05
Yes	Yes	Yes	No	Yes	Yes	No	No	0.95	0.05
Yes	Yes	Yes	No	Yes	No	Yes	Yes	0.95	0.05
Yes	Yes	Yes	No	Yes	No	Yes	No	0.95	0.05
Yes	Yes	Yes	No	Yes	No	No	Yes	0.95	0.05
Yes	Yes	Yes	No	Yes	No	No	No	0.95	0.05
Yes	Yes	Yes	No	No	Yes	Yes	Yes	0.95	0.05
Yes	Yes	Yes	No	No	Yes	Yes	No	0.95	0.05
Yes	Yes	Yes	No	No	Yes	No	Yes	0.95	0.05
Yes	Yes	Yes	No	No	Yes	No	No	0.95	0.05
Yes	Yes	Yes	No	No	No	Yes	Yes	0.95	0.05
Yes	Yes	Yes	No	No	No	Yes	No	0.95	0.05
Yes	Yes	Yes	No	No	No	No	Yes	0.95	0.05
Yes	Yes	Yes	No	No	No	No	No	0.95	0.05
Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	1	0
Yes	Yes	No	Yes	Yes	Yes	Yes	No	1	0
Yes	Yes	No	Yes	Yes	Yes	No	Yes	1	0
Yes	Yes	No	Yes	Yes	Yes	No	No	1	0
Yes	Yes	No	Yes	Yes	No	Yes	Yes	1	0
Yes	Yes	No	Yes	Yes	No	Yes	No	1	0
Yes	Yes	No	Yes	Yes	No	No	Yes	1	0
Yes	Yes	No	Yes	Yes	No	No	No	1	0
Yes	Yes	No	Yes	No	Yes	Yes	Yes	1	0
Yes	Yes	No	Yes	No	Yes	Yes	No	1	0
Yes	Yes	No	Yes	No	Yes	No	Yes	1	0
Yes	Yes	No	Yes	No	Yes	No	No	1	0
Yes	Yes	No	Yes	No	No	Yes	Yes	1	0
Yes	Yes	No	Yes	No	No	Yes	No	1	0
Yes	Yes	No	Yes	No	No	No	Yes	1	0
Yes	Yes	No	Yes	No	No	No	No	1	0
Yes	Yes	No	No	Yes	Yes	Yes	Yes	1	0
Yes	Yes	No	No	Yes	Yes	Yes	No	1	0
Yes	Yes	No	No	Yes	Yes	No	Yes	1	0
Yes	Yes	No	No	Yes	Yes	No	No	1	0
Yes	Yes	No	No	Yes	No	Yes	Yes	1	0
Yes	Yes	No	No	Yes	No	Yes	No	1	0
Yes	Yes	No	No	Yes	No	No	Yes	1	0
Yes	Yes	No	No	Yes	No	No	No	1	0
Yes	Yes	No	No	No	Yes	Yes	Yes	1	0
Yes	Yes	No	No	No	Yes	Yes	No	1	0
Yes	Yes	No	No	No	Yes	No	Yes	1	0
Yes	Yes	No	No	No	Yes	No	No	1	0
Yes	Yes	No	No	No	No	Yes	Yes	1	0
Yes	Yes	No	No	No	No	Yes	No	1	0
Yes	Yes	No	No	No	No	No	Yes	1	0
Yes	Yes	No	No	No	No	No	No	1	0

								Principle 3	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	1	0
Yes	No	Yes	Yes	Yes	Yes	Yes	No	1	0
Yes	No	Yes	Yes	Yes	Yes	No	Yes	1	0
Yes	No	Yes	Yes	Yes	Yes	No	No	1	0
Yes	No	Yes	Yes	Yes	No	Yes	Yes	1	0
Yes	No	Yes	Yes	Yes	No	Yes	No	1	0
Yes	No	Yes	Yes	Yes	No	No	Yes	1	0
Yes	No	Yes	Yes	Yes	No	No	No	1	0
Yes	No	Yes	Yes	No	Yes	Yes	Yes	1	0
Yes	No	Yes	Yes	No	Yes	Yes	No	1	0
Yes	No	Yes	Yes	No	Yes	No	Yes	1	0
Yes	No	Yes	Yes	No	Yes	No	No	1	0
Yes	No	Yes	Yes	No	No	Yes	Yes	1	0
Yes	No	Yes	Yes	No	No	Yes	No	1	0
Yes	No	Yes	Yes	No	No	No	Yes	1	0
Yes	No	Yes	Yes	No	No	No	No	1	0
Yes	No	Yes	No	Yes	Yes	Yes	Yes	1	0
Yes	No	Yes	No	Yes	Yes	Yes	No	1	0
Yes	No	Yes	No	Yes	Yes	No	Yes	1	0
Yes	No	Yes	No	Yes	Yes	No	No	1	0
Yes	No	Yes	No	Yes	No	Yes	Yes	1	0
Yes	No	Yes	No	Yes	No	Yes	No	1	0
Yes	No	Yes	No	Yes	No	No	Yes	1	0
Yes	No	Yes	No	Yes	No	No	No	1	0
Yes	No	Yes	No	No	Yes	Yes	Yes	1	0
Yes	No	Yes	No	No	Yes	Yes	No	1	0
Yes	No	Yes	No	No	Yes	No	Yes	1	0
Yes	No	Yes	No	No	Yes	No	No	1	0
Yes	No	Yes	No	No	No	Yes	Yes	1	0
Yes	No	Yes	No	No	No	Yes	No	1	0
Yes	No	Yes	No	No	No	No	Yes	1	0
Yes	No	Yes	No	No	No	No	No	1	0
Yes	No	No	Yes	Yes	Yes	Yes	Yes	0	1
Yes	No	No	Yes	Yes	Yes	Yes	No	0	1
Yes	No	No	Yes	Yes	Yes	No	Yes	0	1
Yes	No	No	Yes	Yes	Yes	No	No	0	1
Yes	No	No	Yes	Yes	No	Yes	Yes	0	1
Yes	No	No	Yes	Yes	No	Yes	No	0	1
Yes	No	No	Yes	Yes	No	No	Yes	0	1
Yes	No	No	Yes	Yes	No	No	No	0	1
Yes	No	No	Yes	No	Yes	Yes	Yes	0	1
Yes	No	No	Yes	No	Yes	Yes	No	0	1
Yes	No	No	Yes	No	Yes	No	Yes	0	1
Yes	No	No	Yes	No	Yes	No	No	0	1
Yes	No	No	Yes	No	No	Yes	Yes	0	1
Yes	No	No	Yes	No	No	Yes	No	0	1

								Principle 3	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
Yes	No	No	Yes	No	No	No	Yes	0	1
Yes	No	No	Yes	No	No	No	No	0	1
Yes	No	No	No	Yes	Yes	Yes	Yes	0	1
Yes	No	No	No	Yes	Yes	Yes	No	0	1
Yes	No	No	No	Yes	Yes	No	Yes	0	1
Yes	No	No	No	Yes	Yes	No	No	0	1
Yes	No	No	No	Yes	No	Yes	Yes	0	1
Yes	No	No	No	Yes	No	Yes	No	0	1
Yes	No	No	No	Yes	No	No	Yes	0	1
Yes	No	No	No	Yes	No	No	No	0	1
Yes	No	No	No	No	Yes	Yes	Yes	0	1
Yes	No	No	No	No	Yes	Yes	No	0	1
Yes	No	No	No	No	Yes	No	Yes	0	1
Yes	No	No	No	No	Yes	No	No	0	1
Yes	No	No	No	No	No	Yes	Yes	0	1
Yes	No	No	No	No	No	Yes	No	0	1
Yes	No	No	No	No	No	No	Yes	0	1
Yes	No	No	No	No	No	No	No	0	1
No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	1	0
No	Yes	Yes	Yes	Yes	Yes	Yes	No	1	0
No	Yes	Yes	Yes	Yes	Yes	No	Yes	1	0
No	Yes	Yes	Yes	Yes	Yes	No	No	1	0
No	Yes	Yes	Yes	Yes	No	Yes	Yes	1	0
No	Yes	Yes	Yes	Yes	No	Yes	No	1	0
No	Yes	Yes	Yes	Yes	No	No	Yes	1	0
No	Yes	Yes	Yes	Yes	No	No	No	1	0
No	Yes	Yes	Yes	No	Yes	Yes	Yes	1	0
No	Yes	Yes	Yes	No	Yes	Yes	No	1	0
No	Yes	Yes	Yes	No	Yes	No	Yes	1	0
No	Yes	Yes	Yes	No	Yes	No	No	1	0
No	Yes	Yes	Yes	No	No	Yes	Yes	1	0
No	Yes	Yes	Yes	No	No	Yes	No	1	0
No	Yes	Yes	Yes	No	No	No	Yes	1	0
No	Yes	Yes	Yes	No	No	No	No	1	0
No	Yes	Yes	No	Yes	Yes	Yes	Yes	1	0
No	Yes	Yes	No	Yes	Yes	Yes	No	1	0
No	Yes	Yes	No	Yes	Yes	No	Yes	1	0
No	Yes	Yes	No	Yes	Yes	No	No	1	0
No	Yes	Yes	No	Yes	No	Yes	Yes	1	0
No	Yes	Yes	No	Yes	No	Yes	No	1	0
No	Yes	Yes	No	Yes	No	No	Yes	1	0
No	Yes	Yes	No	Yes	No	No	No	1	0
No	Yes	Yes	No	No	Yes	Yes	Yes	1	0
No	Yes	Yes	No	No	Yes	Yes	No	1	0
No	Yes	Yes	No	No	Yes	No	Yes	1	0
No	Yes	Yes	No	No	Yes	No	No	1	0

								Principle 3	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
No	Yes	Yes	No	No	No	Yes	Yes	1	0
No	Yes	Yes	No	No	No	Yes	No	1	0
No	Yes	Yes	No	No	No	No	Yes	1	0
No	Yes	Yes	No	No	No	No	No	1	0
No	Yes	No	Yes	Yes	Yes	Yes	Yes	1	0
No	Yes	No	Yes	Yes	Yes	Yes	No	1	0
No	Yes	No	Yes	Yes	Yes	No	Yes	1	0
No	Yes	No	Yes	Yes	Yes	No	No	1	0
No	Yes	No	Yes	Yes	No	Yes	Yes	1	0
No	Yes	No	Yes	Yes	No	Yes	No	1	0
No	Yes	No	Yes	Yes	No	No	Yes	1	0
No	Yes	No	Yes	Yes	No	No	No	1	0
No	Yes	No	Yes	No	Yes	Yes	Yes	1	0
No	Yes	No	Yes	No	Yes	Yes	No	1	0
No	Yes	No	Yes	No	Yes	No	Yes	1	0
No	Yes	No	Yes	No	Yes	No	No	1	0
No	Yes	No	Yes	No	No	Yes	Yes	1	0
No	Yes	No	Yes	No	No	Yes	No	1	0
No	Yes	No	Yes	No	No	No	Yes	1	0
No	Yes	No	Yes	No	No	No	No	1	0
No	Yes	No	No	Yes	Yes	Yes	Yes	1	0
No	Yes	No	No	Yes	Yes	Yes	No	1	0
No	Yes	No	No	Yes	Yes	No	Yes	1	0
No	Yes	No	No	Yes	Yes	No	No	1	0
No	Yes	No	No	Yes	No	Yes	Yes	1	0
No	Yes	No	No	Yes	No	Yes	No	1	0
No	Yes	No	No	Yes	No	No	Yes	1	0
No	Yes	No	No	Yes	No	No	No	1	0
No	Yes	No	No	No	Yes	Yes	Yes	1	0
No	Yes	No	No	No	Yes	Yes	No	1	0
No	Yes	No	No	No	Yes	No	Yes	1	0
No	Yes	No	No	No	Yes	No	No	1	0
No	Yes	No	No	No	No	Yes	Yes	1	0
No	Yes	No	No	No	No	Yes	No	1	0
No	Yes	No	No	No	No	No	Yes	1	0
No	Yes	No	No	No	No	No	No	0	1
No	No	Yes	Yes	Yes	Yes	Yes	Yes	1	0
No	No	Yes	Yes	Yes	Yes	Yes	No	1	0
No	No	Yes	Yes	Yes	Yes	No	Yes	1	0
No	No	Yes	Yes	Yes	Yes	No	No	1	0
No	No	Yes	Yes	Yes	No	Yes	Yes	1	0
No	No	Yes	Yes	Yes	No	Yes	No	1	0
No	No	Yes	Yes	Yes	No	No	Yes	1	0
No	No	Yes	Yes	Yes	No	No	No	1	0
No	No	Yes	Yes	No	Yes	Yes	Yes	1	0
No	No	Yes	Yes	No	Yes	Yes	No	1	0

								Principle 3	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
No	No	Yes	Yes	No	Yes	No	Yes	1	0
No	No	Yes	Yes	No	Yes	No	No	1	0
No	No	Yes	Yes	No	No	Yes	Yes	1	0
No	No	Yes	Yes	No	No	Yes	No	1	0
No	No	Yes	Yes	No	No	No	Yes	1	0
No	No	Yes	Yes	No	No	No	No	1	0
No	No	Yes	No	Yes	Yes	Yes	Yes	1	0
No	No	Yes	No	Yes	Yes	Yes	No	1	0
No	No	Yes	No	Yes	Yes	No	Yes	1	0
No	No	Yes	No	Yes	Yes	No	No	1	0
No	No	Yes	No	Yes	No	Yes	Yes	1	0
No	No	Yes	No	Yes	No	Yes	No	1	0
No	No	Yes	No	Yes	No	No	Yes	1	0
No	No	Yes	No	Yes	No	No	No	1	0
No	No	Yes	No	No	Yes	Yes	Yes	1	0
No	No	Yes	No	No	Yes	Yes	No	1	0
No	No	Yes	No	No	Yes	No	Yes	1	0
No	No	Yes	No	No	Yes	No	No	1	0
No	No	Yes	No	No	No	Yes	Yes	1	0
No	No	Yes	No	No	No	Yes	No	1	0
No	No	Yes	No	No	No	No	Yes	1	0
No	No	Yes	No	No	No	No	No	0	1
No	No	No	Yes	Yes	Yes	Yes	Yes	0	1
No	No	No	Yes	Yes	Yes	Yes	No	0	1
No	No	No	Yes	Yes	Yes	No	Yes	0	1
No	No	No	Yes	Yes	Yes	No	No	0	1
No	No	No	Yes	Yes	No	Yes	Yes	0	1
No	No	No	Yes	Yes	No	Yes	No	0	1
No	No	No	Yes	Yes	No	No	Yes	0	1
No	No	No	Yes	Yes	No	No	No	0	1
No	No	No	Yes	No	Yes	Yes	Yes	0	1
No	No	No	Yes	No	Yes	Yes	No	0	1
No	No	No	Yes	No	Yes	No	Yes	0	1
No	No	No	Yes	No	Yes	No	No	0	1
No	No	No	Yes	No	No	Yes	Yes	0	1
No	No	No	Yes	No	No	Yes	No	0	1
No	No	No	Yes	No	No	No	Yes	0	1
No	No	No	Yes	No	No	No	No	0	1
No	No	No	No	Yes	Yes	Yes	Yes	0	1
No	No	No	No	Yes	Yes	Yes	No	0	1
No	No	No	No	Yes	Yes	No	Yes	0	1
No	No	No	No	Yes	Yes	No	No	0	1
No	No	No	No	Yes	No	Yes	Yes	0	1
No	No	No	No	Yes	No	Yes	No	0	1
No	No	No	No	Yes	No	No	Yes	0	1
No	No	No	No	Yes	No	No	No	0	1

								Principle 3	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
No	No	No	No	No	Yes	Yes	Yes	0	1
No	No	No	No	No	Yes	Yes	No	0	1
No	No	No	No	No	Yes	No	Yes	0	1
No	No	No	No	No	Yes	No	No	0	1
No	No	No	No	No	No	Yes	Yes	0	1
No	No	No	No	No	No	Yes	No	0	1
No	No	No	No	No	No	No	Yes	0	1
No	No	No	No	No	No	No	No	0	1

Table A3-34. Priciple4 Node Conditional Probability Table

								Principle 4	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	1	0
Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	1	0
Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	1	0
Yes	Yes	Yes	Yes	Yes	Yes	No	No	1	0
Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	1	0
Yes	Yes	Yes	Yes	Yes	No	Yes	No	1	0
Yes	Yes	Yes	Yes	Yes	No	No	Yes	1	0
Yes	Yes	Yes	Yes	Yes	No	No	No	1	0
Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	1	0
Yes	Yes	Yes	Yes	No	Yes	Yes	No	1	0
Yes	Yes	Yes	Yes	No	Yes	No	Yes	1	0
Yes	Yes	Yes	Yes	No	Yes	No	No	1	0
Yes	Yes	Yes	Yes	No	No	Yes	Yes	1	0
Yes	Yes	Yes	Yes	No	No	Yes	No	1	0
Yes	Yes	Yes	Yes	No	No	No	Yes	1	0
Yes	Yes	Yes	Yes	No	No	No	No	1	0
Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	1	0
Yes	Yes	Yes	No	Yes	Yes	Yes	No	1	0
Yes	Yes	Yes	No	Yes	Yes	No	Yes	1	0
Yes	Yes	Yes	No	Yes	Yes	No	No	1	0
Yes	Yes	Yes	No	Yes	No	Yes	Yes	1	0
Yes	Yes	Yes	No	Yes	No	Yes	No	1	0
Yes	Yes	Yes	No	Yes	No	No	Yes	1	0
Yes	Yes	Yes	No	Yes	No	No	No	1	0
Yes	Yes	Yes	No	No	Yes	Yes	Yes	1	0
Yes	Yes	Yes	No	No	Yes	Yes	No	1	0
Yes	Yes	Yes	No	No	Yes	No	Yes	1	0
Yes	Yes	Yes	No	No	Yes	No	No	1	0
Yes	Yes	Yes	No	No	No	Yes	Yes	1	0
Yes	Yes	Yes	No	No	No	Yes	No	1	0
Yes	Yes	Yes	No	No	No	No	Yes	1	0
Yes	Yes	Yes	No	No	No	No	No	1	0

								Principle 4	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	1	0
Yes	Yes	No	Yes	Yes	Yes	Yes	No	1	0
Yes	Yes	No	Yes	Yes	Yes	No	Yes	1	0
Yes	Yes	No	Yes	Yes	Yes	No	No	1	0
Yes	Yes	No	Yes	Yes	No	Yes	Yes	1	0
Yes	Yes	No	Yes	Yes	No	Yes	No	1	0
Yes	Yes	No	Yes	Yes	No	No	Yes	1	0
Yes	Yes	No	Yes	Yes	No	No	No	1	0
Yes	Yes	No	Yes	No	Yes	Yes	Yes	1	0
Yes	Yes	No	Yes	No	Yes	Yes	No	1	0
Yes	Yes	No	Yes	No	Yes	No	Yes	1	0
Yes	Yes	No	Yes	No	Yes	No	No	1	0
Yes	Yes	No	Yes	No	No	Yes	Yes	1	0
Yes	Yes	No	Yes	No	No	Yes	No	1	0
Yes	Yes	No	Yes	No	No	No	Yes	1	0
Yes	Yes	No	Yes	No	No	No	No	1	0
Yes	Yes	No	No	Yes	Yes	Yes	Yes	1	0
Yes	Yes	No	No	Yes	Yes	Yes	No	1	0
Yes	Yes	No	No	Yes	Yes	No	Yes	1	0
Yes	Yes	No	No	Yes	Yes	No	No	1	0
Yes	Yes	No	No	Yes	No	Yes	Yes	1	0
Yes	Yes	No	No	Yes	No	Yes	No	1	0
Yes	Yes	No	No	Yes	No	No	Yes	1	0
Yes	Yes	No	No	Yes	No	No	No	1	0
Yes	Yes	No	No	No	Yes	Yes	Yes	1	0
Yes	Yes	No	No	No	Yes	Yes	No	1	0
Yes	Yes	No	No	No	Yes	No	Yes	1	0
Yes	Yes	No	No	No	Yes	No	No	1	0
Yes	Yes	No	No	No	No	Yes	Yes	1	0
Yes	Yes	No	No	No	No	Yes	No	1	0
Yes	Yes	No	No	No	No	No	Yes	1	0
Yes	Yes	No	No	No	No	No	No	1	0
Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	1	0
Yes	No	Yes	Yes	Yes	Yes	Yes	No	1	0
Yes	No	Yes	Yes	Yes	Yes	No	Yes	1	0
Yes	No	Yes	Yes	Yes	Yes	No	No	1	0
Yes	No	Yes	Yes	Yes	No	Yes	Yes	1	0
Yes	No	Yes	Yes	Yes	No	Yes	No	1	0
Yes	No	Yes	Yes	Yes	No	No	Yes	1	0
Yes	No	Yes	Yes	Yes	No	No	No	1	0
Yes	No	Yes	Yes	No	Yes	Yes	Yes	1	0
Yes	No	Yes	Yes	No	Yes	Yes	No	1	0
Yes	No	Yes	Yes	No	Yes	No	Yes	1	0
Yes	No	Yes	Yes	No	Yes	No	No	1	0
Yes	No	Yes	Yes	No	No	Yes	Yes	1	0
Yes	No	Yes	Yes	No	No	Yes	No	1	0

								Principle 4	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
Yes	No	Yes	Yes	No	No	No	Yes	1	0
Yes	No	Yes	Yes	No	No	No	No	1	0
Yes	No	Yes	No	Yes	Yes	Yes	Yes	1	0
Yes	No	Yes	No	Yes	Yes	Yes	No	1	0
Yes	No	Yes	No	Yes	Yes	No	Yes	1	0
Yes	No	Yes	No	Yes	Yes	No	No	1	0
Yes	No	Yes	No	Yes	No	Yes	Yes	1	0
Yes	No	Yes	No	Yes	No	Yes	No	1	0
Yes	No	Yes	No	Yes	No	No	Yes	1	0
Yes	No	Yes	No	Yes	No	No	No	1	0
Yes	No	Yes	No	No	Yes	Yes	Yes	1	0
Yes	No	Yes	No	No	Yes	Yes	No	1	0
Yes	No	Yes	No	No	Yes	No	Yes	1	0
Yes	No	Yes	No	No	Yes	No	No	1	0
Yes	No	Yes	No	No	No	Yes	Yes	1	0
Yes	No	Yes	No	No	No	Yes	No	1	0
Yes	No	Yes	No	No	No	No	Yes	1	0
Yes	No	Yes	No	No	No	No	No	1	0
Yes	No	No	Yes	Yes	Yes	Yes	Yes	1	0
Yes	No	No	Yes	Yes	Yes	Yes	No	1	0
Yes	No	No	Yes	Yes	Yes	No	Yes	1	0
Yes	No	No	Yes	Yes	Yes	No	No	1	0
Yes	No	No	Yes	Yes	No	Yes	Yes	0	1
Yes	No	No	Yes	Yes	No	Yes	No	0	1
Yes	No	No	Yes	Yes	No	No	Yes	0	1
Yes	No	No	Yes	Yes	No	No	No	0	1
Yes	No	No	Yes	No	Yes	Yes	Yes	1	0
Yes	No	No	Yes	No	Yes	Yes	No	1	0
Yes	No	No	Yes	No	Yes	No	Yes	1	0
Yes	No	No	Yes	No	Yes	No	No	1	0
Yes	No	No	Yes	No	No	Yes	Yes	0	1
Yes	No	No	Yes	No	No	Yes	No	0	1
Yes	No	No	Yes	No	No	No	Yes	0	1
Yes	No	No	Yes	No	No	No	No	0	1
Yes	No	No	No	Yes	Yes	Yes	Yes	1	0
Yes	No	No	No	Yes	Yes	Yes	No	1	0
Yes	No	No	No	Yes	Yes	No	Yes	1	0
Yes	No	No	No	Yes	Yes	No	No	1	0
Yes	No	No	No	Yes	No	Yes	Yes	0	1
Yes	No	No	No	Yes	No	Yes	No	0	1
Yes	No	No	No	Yes	No	No	Yes	0	1
Yes	No	No	No	Yes	No	No	No	0	1
Yes	No	No	No	No	Yes	Yes	Yes	1	0
Yes	No	No	No	No	Yes	Yes	No	1	0
Yes	No	No	No	No	Yes	No	Yes	1	0
Yes	No	No	No	No	Yes	No	No	1	0

Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Principle 4	
								Yes	No
Yes	No	No	No	No	No	Yes	Yes	0	1
Yes	No	No	No	No	No	Yes	No	0	1
Yes	No	No	No	No	No	No	Yes	0	1
Yes	No	No	No	No	No	No	No	0	1
No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	1	0
No	Yes	Yes	Yes	Yes	Yes	Yes	No	1	0
No	Yes	Yes	Yes	Yes	Yes	No	Yes	1	0
No	Yes	Yes	Yes	Yes	Yes	No	No	1	0
No	Yes	Yes	Yes	Yes	No	Yes	Yes	1	0
No	Yes	Yes	Yes	Yes	No	Yes	No	1	0
No	Yes	Yes	Yes	Yes	No	No	Yes	1	0
No	Yes	Yes	Yes	Yes	No	No	No	1	0
No	Yes	Yes	Yes	No	Yes	Yes	Yes	1	0
No	Yes	Yes	Yes	No	Yes	Yes	No	1	0
No	Yes	Yes	Yes	No	Yes	No	Yes	1	0
No	Yes	Yes	Yes	No	Yes	No	No	1	0
No	Yes	Yes	Yes	No	No	Yes	Yes	1	0
No	Yes	Yes	Yes	No	No	Yes	No	1	0
No	Yes	Yes	Yes	No	No	No	Yes	1	0
No	Yes	Yes	Yes	No	No	No	No	1	0
No	Yes	Yes	No	Yes	Yes	Yes	Yes	1	0
No	Yes	Yes	No	Yes	Yes	Yes	No	1	0
No	Yes	Yes	No	Yes	Yes	No	Yes	1	0
No	Yes	Yes	No	Yes	Yes	No	No	1	0
No	Yes	Yes	No	Yes	No	Yes	Yes	1	0
No	Yes	Yes	No	Yes	No	Yes	No	1	0
No	Yes	Yes	No	Yes	No	No	Yes	1	0
No	Yes	Yes	No	Yes	No	No	No	1	0
No	Yes	Yes	No	No	Yes	Yes	Yes	1	0
No	Yes	Yes	No	No	Yes	Yes	No	1	0
No	Yes	Yes	No	No	Yes	No	Yes	1	0
No	Yes	Yes	No	No	Yes	No	No	1	0
No	Yes	Yes	No	No	No	Yes	Yes	1	0
No	Yes	Yes	No	No	No	Yes	No	1	0
No	Yes	Yes	No	No	No	No	Yes	1	0
No	Yes	Yes	No	No	No	No	No	1	0
No	Yes	No	Yes	Yes	Yes	Yes	Yes	1	0
No	Yes	No	Yes	Yes	Yes	Yes	No	1	0
No	Yes	No	Yes	Yes	Yes	No	Yes	1	0
No	Yes	No	Yes	Yes	Yes	No	No	1	0
No	Yes	No	Yes	Yes	No	Yes	Yes	1	0
No	Yes	No	Yes	Yes	No	Yes	No	1	0
No	Yes	No	Yes	Yes	No	No	Yes	1	0
No	Yes	No	Yes	Yes	No	No	No	1	0
No	Yes	No	Yes	No	Yes	Yes	Yes	1	0
No	Yes	No	Yes	No	Yes	Yes	No	1	0

								Principle 4	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
No	Yes	No	Yes	No	Yes	No	Yes	1	0
No	Yes	No	Yes	No	Yes	No	No	1	0
No	Yes	No	Yes	No	No	Yes	Yes	1	0
No	Yes	No	Yes	No	No	Yes	No	1	0
No	Yes	No	Yes	No	No	No	Yes	1	0
No	Yes	No	Yes	No	No	No	No	1	0
No	Yes	No	No	Yes	Yes	Yes	Yes	1	0
No	Yes	No	No	Yes	Yes	Yes	No	1	0
No	Yes	No	No	Yes	Yes	No	Yes	1	0
No	Yes	No	No	Yes	Yes	No	No	1	0
No	Yes	No	No	Yes	No	Yes	Yes	1	0
No	Yes	No	No	Yes	No	Yes	No	1	0
No	Yes	No	No	Yes	No	No	Yes	1	0
No	Yes	No	No	Yes	No	No	No	1	0
No	Yes	No	No	No	Yes	Yes	Yes	1	0
No	Yes	No	No	No	Yes	Yes	No	1	0
No	Yes	No	No	No	Yes	No	Yes	1	0
No	Yes	No	No	No	Yes	No	No	1	0
No	Yes	No	No	No	No	Yes	Yes	1	0
No	Yes	No	No	No	No	Yes	No	1	0
No	Yes	No	No	No	No	No	Yes	1	0
No	Yes	No	No	No	No	No	No	0	1
No	No	Yes	Yes	Yes	Yes	Yes	Yes	1	0
No	No	Yes	Yes	Yes	Yes	Yes	No	1	0
No	No	Yes	Yes	Yes	Yes	No	Yes	1	0
No	No	Yes	Yes	Yes	Yes	No	No	1	0
No	No	Yes	Yes	Yes	No	Yes	Yes	1	0
No	No	Yes	Yes	Yes	No	Yes	No	1	0
No	No	Yes	Yes	Yes	No	No	Yes	1	0
No	No	Yes	Yes	Yes	No	No	No	1	0
No	No	Yes	Yes	No	Yes	Yes	Yes	1	0
No	No	Yes	Yes	No	Yes	Yes	No	1	0
No	No	Yes	Yes	No	Yes	No	Yes	1	0
No	No	Yes	Yes	No	Yes	No	No	1	0
No	No	Yes	Yes	No	No	Yes	Yes	1	0
No	No	Yes	Yes	No	No	Yes	No	1	0
No	No	Yes	Yes	No	No	No	Yes	1	0
No	No	Yes	Yes	No	No	No	No	1	0
No	No	Yes	No	Yes	Yes	Yes	Yes	1	0
No	No	Yes	No	Yes	Yes	Yes	No	1	0
No	No	Yes	No	Yes	Yes	No	Yes	1	0
No	No	Yes	No	Yes	Yes	No	No	1	0
No	No	Yes	No	Yes	No	Yes	Yes	1	0
No	No	Yes	No	Yes	No	Yes	No	1	0
No	No	Yes	No	Yes	No	No	Yes	1	0
No	No	Yes	No	Yes	No	No	No	1	0

								Principle 4	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
No	No	Yes	No	No	Yes	Yes	Yes	1	0
No	No	Yes	No	No	Yes	Yes	No	1	0
No	No	Yes	No	No	Yes	No	Yes	1	0
No	No	Yes	No	No	Yes	No	No	1	0
No	No	Yes	No	No	No	Yes	Yes	1	0
No	No	Yes	No	No	No	Yes	No	1	0
No	No	Yes	No	No	No	No	Yes	1	0
No	No	Yes	No	No	No	No	No	0	1
No	No	No	Yes	Yes	Yes	Yes	Yes	1	0
No	No	No	Yes	Yes	Yes	Yes	No	1	0
No	No	No	Yes	Yes	Yes	No	Yes	1	0
No	No	No	Yes	Yes	Yes	No	No	1	0
No	No	No	Yes	Yes	No	Yes	Yes	0	1
No	No	No	Yes	Yes	No	Yes	No	0	1
No	No	No	Yes	Yes	No	No	Yes	0	1
No	No	No	Yes	Yes	No	No	No	0	1
No	No	No	Yes	No	Yes	Yes	Yes	1	0
No	No	No	Yes	No	Yes	Yes	No	1	0
No	No	No	Yes	No	Yes	No	Yes	1	0
No	No	No	Yes	No	Yes	No	No	1	0
No	No	No	Yes	No	No	Yes	Yes	0	1
No	No	No	Yes	No	No	Yes	No	0	1
No	No	No	Yes	No	No	No	Yes	0	1
No	No	No	Yes	No	No	No	No	0	1
No	No	No	No	Yes	Yes	Yes	Yes	1	0
No	No	No	No	Yes	Yes	Yes	No	1	0
No	No	No	No	Yes	Yes	No	Yes	1	0
No	No	No	No	Yes	Yes	No	No	1	0
No	No	No	No	Yes	No	Yes	Yes	0	1
No	No	No	No	Yes	No	Yes	No	0	1
No	No	No	No	Yes	No	No	Yes	0	1
No	No	No	No	Yes	No	No	No	0	1
No	No	No	No	No	Yes	Yes	Yes	1	0
No	No	No	No	No	Yes	Yes	No	1	0
No	No	No	No	No	Yes	No	Yes	1	0
No	No	No	No	No	Yes	No	No	0	1
No	No	No	No	No	No	Yes	Yes	0	1
No	No	No	No	No	No	Yes	No	0	1
No	No	No	No	No	No	No	Yes	0	1
No	No	No	No	No	No	No	No	0	1

Table A3-35. Priciple5 Node Conditional Probability Table

								Principle 5	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	1	0
Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	1	0
Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	1	0
Yes	Yes	Yes	Yes	Yes	Yes	No	No	1	0
Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	1	0
Yes	Yes	Yes	Yes	Yes	No	Yes	No	1	0
Yes	Yes	Yes	Yes	Yes	No	No	Yes	1	0
Yes	Yes	Yes	Yes	Yes	No	No	No	1	0
Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	1	0
Yes	Yes	Yes	Yes	No	Yes	Yes	No	1	0
Yes	Yes	Yes	Yes	No	Yes	No	Yes	0	1
Yes	Yes	Yes	Yes	No	Yes	No	No	0	1
Yes	Yes	Yes	Yes	No	No	Yes	Yes	1	0
Yes	Yes	Yes	Yes	No	No	Yes	No	1	0
Yes	Yes	Yes	Yes	No	No	No	Yes	0	1
Yes	Yes	Yes	Yes	No	No	No	No	0	1
Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	1	0
Yes	Yes	Yes	No	Yes	Yes	Yes	No	1	0
Yes	Yes	Yes	No	Yes	Yes	No	Yes	1	0
Yes	Yes	Yes	No	Yes	Yes	No	No	1	0
Yes	Yes	Yes	No	Yes	No	Yes	Yes	1	0
Yes	Yes	Yes	No	Yes	No	Yes	No	1	0
Yes	Yes	Yes	No	Yes	No	No	Yes	1	0
Yes	Yes	Yes	No	Yes	No	No	No	1	0
Yes	Yes	Yes	No	No	Yes	Yes	Yes	1	0
Yes	Yes	Yes	No	No	Yes	Yes	No	1	0
Yes	Yes	Yes	No	No	Yes	No	Yes	0	1
Yes	Yes	Yes	No	No	Yes	No	No	0	1
Yes	Yes	Yes	No	No	No	Yes	Yes	1	0
Yes	Yes	Yes	No	No	No	Yes	No	1	0
Yes	Yes	Yes	No	No	No	No	Yes	0	1
Yes	Yes	Yes	No	No	No	No	No	0	1
Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	1	0
Yes	Yes	No	Yes	Yes	Yes	Yes	No	1	0
Yes	Yes	No	Yes	Yes	Yes	No	Yes	1	0
Yes	Yes	No	Yes	Yes	Yes	No	No	1	0
Yes	Yes	No	Yes	Yes	No	Yes	Yes	1	0
Yes	Yes	No	Yes	Yes	No	Yes	No	1	0
Yes	Yes	No	Yes	Yes	No	No	Yes	1	0
Yes	Yes	No	Yes	Yes	No	No	No	1	0
Yes	Yes	No	Yes	No	Yes	Yes	Yes	1	0
Yes	Yes	No	Yes	No	Yes	Yes	No	1	0
Yes	Yes	No	Yes	No	Yes	No	Yes	0	1
Yes	Yes	No	Yes	No	Yes	No	No	0	1

								Principle 5	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
Yes	Yes	No	Yes	No	No	Yes	Yes	1	0
Yes	Yes	No	Yes	No	No	Yes	No	1	0
Yes	Yes	No	Yes	No	No	No	Yes	0	1
Yes	Yes	No	Yes	No	No	No	No	0	1
Yes	Yes	No	No	Yes	Yes	Yes	Yes	1	0
Yes	Yes	No	No	Yes	Yes	Yes	No	1	0
Yes	Yes	No	No	Yes	Yes	No	Yes	1	0
Yes	Yes	No	No	Yes	Yes	No	No	1	0
Yes	Yes	No	No	Yes	No	Yes	Yes	1	0
Yes	Yes	No	No	Yes	No	Yes	No	1	0
Yes	Yes	No	No	Yes	No	No	Yes	1	0
Yes	Yes	No	No	Yes	No	No	No	1	0
Yes	Yes	No	No	No	Yes	Yes	Yes	1	0
Yes	Yes	No	No	No	Yes	Yes	No	1	0
Yes	Yes	No	No	No	Yes	No	Yes	0	1
Yes	Yes	No	No	No	Yes	No	No	0	1
Yes	Yes	No	No	No	No	Yes	Yes	1	0
Yes	Yes	No	No	No	No	Yes	No	1	0
Yes	Yes	No	No	No	No	No	Yes	0	1
Yes	Yes	No	No	No	No	No	No	0	1
Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	1	0
Yes	No	Yes	Yes	Yes	Yes	Yes	No	1	0
Yes	No	Yes	Yes	Yes	Yes	No	Yes	1	0
Yes	No	Yes	Yes	Yes	Yes	No	No	1	0
Yes	No	Yes	Yes	Yes	No	Yes	Yes	1	0
Yes	No	Yes	Yes	Yes	No	Yes	No	1	0
Yes	No	Yes	Yes	Yes	No	No	Yes	1	0
Yes	No	Yes	Yes	Yes	No	No	No	1	0
Yes	No	Yes	Yes	No	Yes	Yes	Yes	1	0
Yes	No	Yes	Yes	No	Yes	Yes	No	1	0
Yes	No	Yes	Yes	No	Yes	No	Yes	0	1
Yes	No	Yes	Yes	No	Yes	No	No	0	1
Yes	No	Yes	Yes	No	No	Yes	Yes	1	0
Yes	No	Yes	Yes	No	No	Yes	No	1	0
Yes	No	Yes	Yes	No	No	No	Yes	0	1
Yes	No	Yes	Yes	No	No	No	No	0	1
Yes	No	Yes	No	Yes	Yes	Yes	Yes	1	0
Yes	No	Yes	No	Yes	Yes	Yes	No	1	0
Yes	No	Yes	No	Yes	Yes	No	Yes	1	0
Yes	No	Yes	No	Yes	Yes	No	No	1	0
Yes	No	Yes	No	Yes	No	Yes	Yes	1	0
Yes	No	Yes	No	Yes	No	Yes	No	1	0
Yes	No	Yes	No	Yes	No	No	Yes	1	0
Yes	No	Yes	No	Yes	No	No	No	1	0
Yes	No	Yes	No	No	Yes	Yes	Yes	1	0
Yes	No	Yes	No	No	Yes	Yes	No	1	0

								Principle 5	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
Yes	No	Yes	No	No	Yes	No	Yes	0	1
Yes	No	Yes	No	No	Yes	No	No	0	1
Yes	No	Yes	No	No	No	Yes	Yes	1	0
Yes	No	Yes	No	No	No	Yes	No	1	0
Yes	No	Yes	No	No	No	No	Yes	0	1
Yes	No	Yes	No	No	No	No	No	0	1
Yes	No	No	Yes	Yes	Yes	Yes	Yes	1	0
Yes	No	No	Yes	Yes	Yes	Yes	No	1	0
Yes	No	No	Yes	Yes	Yes	No	Yes	1	0
Yes	No	No	Yes	Yes	Yes	No	No	1	0
Yes	No	No	Yes	Yes	No	Yes	Yes	1	0
Yes	No	No	Yes	Yes	No	Yes	No	1	0
Yes	No	No	Yes	Yes	No	No	Yes	1	0
Yes	No	No	Yes	Yes	No	No	No	1	0
Yes	No	No	Yes	No	Yes	Yes	Yes	1	0
Yes	No	No	Yes	No	Yes	Yes	No	1	0
Yes	No	No	Yes	No	Yes	No	Yes	0	1
Yes	No	No	Yes	No	Yes	No	No	0	1
Yes	No	No	Yes	No	No	Yes	Yes	1	0
Yes	No	No	Yes	No	No	Yes	No	1	0
Yes	No	No	Yes	No	No	No	Yes	0	1
Yes	No	No	Yes	No	No	No	No	0	1
Yes	No	No	No	Yes	Yes	Yes	Yes	1	0
Yes	No	No	No	Yes	Yes	Yes	No	1	0
Yes	No	No	No	Yes	Yes	No	Yes	1	0
Yes	No	No	No	Yes	Yes	No	No	1	0
Yes	No	No	No	Yes	No	Yes	Yes	1	0
Yes	No	No	No	Yes	No	Yes	No	1	0
Yes	No	No	No	Yes	No	No	Yes	1	0
Yes	No	No	No	Yes	No	No	No	1	0
Yes	No	No	No	No	Yes	Yes	Yes	1	0
Yes	No	No	No	No	Yes	Yes	No	1	0
Yes	No	No	No	No	Yes	No	Yes	0	1
Yes	No	No	No	No	Yes	No	No	0	1
Yes	No	No	No	No	No	Yes	Yes	1	0
Yes	No	No	No	No	No	Yes	No	1	0
Yes	No	No	No	No	No	No	Yes	0	1
Yes	No	No	No	No	No	No	No	0	1
No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	1	0
No	Yes	Yes	Yes	Yes	Yes	Yes	No	1	0
No	Yes	Yes	Yes	Yes	Yes	No	Yes	1	0
No	Yes	Yes	Yes	Yes	Yes	No	No	1	0
No	Yes	Yes	Yes	Yes	No	Yes	Yes	1	0
No	Yes	Yes	Yes	Yes	No	Yes	No	1	0
No	Yes	Yes	Yes	Yes	No	No	Yes	1	0
No	Yes	Yes	Yes	Yes	No	No	No	1	0

Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Principle 5	
								Yes	No
No	Yes	Yes	Yes	No	Yes	Yes	Yes	1	0
No	Yes	Yes	Yes	No	Yes	Yes	No	1	0
No	Yes	Yes	Yes	No	Yes	No	Yes	0	1
No	Yes	Yes	Yes	No	Yes	No	No	0	1
No	Yes	Yes	Yes	No	No	Yes	Yes	1	0
No	Yes	Yes	Yes	No	No	Yes	No	1	0
No	Yes	Yes	Yes	No	No	No	Yes	0	1
No	Yes	Yes	Yes	No	No	No	No	0	1
No	Yes	Yes	No	Yes	Yes	Yes	Yes	1	0
No	Yes	Yes	No	Yes	Yes	Yes	No	1	0
No	Yes	Yes	No	Yes	Yes	No	Yes	1	0
No	Yes	Yes	No	Yes	Yes	No	No	1	0
No	Yes	Yes	No	Yes	No	Yes	Yes	1	0
No	Yes	Yes	No	Yes	No	Yes	No	1	0
No	Yes	Yes	No	Yes	No	No	Yes	1	0
No	Yes	Yes	No	Yes	No	No	No	1	0
No	Yes	Yes	No	No	Yes	Yes	Yes	1	0
No	Yes	Yes	No	No	Yes	Yes	No	1	0
No	Yes	Yes	No	No	Yes	No	Yes	0	1
No	Yes	Yes	No	No	Yes	No	No	0	1
No	Yes	Yes	No	No	No	Yes	Yes	1	0
No	Yes	Yes	No	No	No	Yes	No	1	0
No	Yes	Yes	No	No	No	No	Yes	0	1
No	Yes	Yes	No	No	No	No	No	0	1
No	Yes	No	Yes	Yes	Yes	Yes	Yes	1	0
No	Yes	No	Yes	Yes	Yes	Yes	No	1	0
No	Yes	No	Yes	Yes	Yes	No	Yes	1	0
No	Yes	No	Yes	Yes	Yes	No	No	1	0
No	Yes	No	Yes	Yes	No	Yes	Yes	1	0
No	Yes	No	Yes	Yes	No	Yes	No	1	0
No	Yes	No	Yes	Yes	No	No	Yes	1	0
No	Yes	No	Yes	Yes	No	No	No	1	0
No	Yes	No	Yes	No	Yes	Yes	Yes	1	0
No	Yes	No	Yes	No	Yes	Yes	No	1	0
No	Yes	No	Yes	No	Yes	No	Yes	0	1
No	Yes	No	Yes	No	Yes	No	No	0	1
No	Yes	No	Yes	No	No	Yes	Yes	1	0
No	Yes	No	Yes	No	No	Yes	No	1	0
No	Yes	No	Yes	No	No	No	Yes	0	1
No	Yes	No	Yes	No	No	No	No	0	1
No	Yes	No	No	Yes	Yes	Yes	Yes	1	0
No	Yes	No	No	Yes	Yes	Yes	No	1	0
No	Yes	No	No	Yes	Yes	No	Yes	1	0
No	Yes	No	No	Yes	Yes	No	No	1	0
No	Yes	No	No	Yes	No	Yes	Yes	1	0
No	Yes	No	No	Yes	No	Yes	No	1	0

								Principle 5	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
No	Yes	No	No	Yes	No	No	Yes	1	0
No	Yes	No	No	Yes	No	No	No	1	0
No	Yes	No	No	No	Yes	Yes	Yes	1	0
No	Yes	No	No	No	Yes	Yes	No	1	0
No	Yes	No	No	No	Yes	No	Yes	0	1
No	Yes	No	No	No	Yes	No	No	0	1
No	Yes	No	No	No	No	Yes	Yes	1	0
No	Yes	No	No	No	No	Yes	No	1	0
No	Yes	No	No	No	No	No	Yes	0	1
No	Yes	No	No	No	No	No	No	0	1
No	No	Yes	Yes	Yes	Yes	Yes	Yes	1	0
No	No	Yes	Yes	Yes	Yes	Yes	No	1	0
No	No	Yes	Yes	Yes	Yes	No	Yes	1	0
No	No	Yes	Yes	Yes	Yes	No	No	1	0
No	No	Yes	Yes	Yes	No	Yes	Yes	1	0
No	No	Yes	Yes	Yes	No	Yes	No	1	0
No	No	Yes	Yes	Yes	No	No	Yes	1	0
No	No	Yes	Yes	Yes	No	No	No	1	0
No	No	Yes	Yes	No	Yes	Yes	Yes	1	0
No	No	Yes	Yes	No	Yes	Yes	No	1	0
No	No	Yes	Yes	No	Yes	No	Yes	0	1
No	No	Yes	Yes	No	Yes	No	No	0	1
No	No	Yes	Yes	No	No	Yes	Yes	1	0
No	No	Yes	Yes	No	No	Yes	No	1	0
No	No	Yes	Yes	No	No	No	Yes	0	1
No	No	Yes	Yes	No	No	No	No	0	1
No	No	Yes	No	Yes	Yes	Yes	Yes	1	0
No	No	Yes	No	Yes	Yes	Yes	No	1	0
No	No	Yes	No	Yes	Yes	No	Yes	1	0
No	No	Yes	No	Yes	Yes	No	No	1	0
No	No	Yes	No	Yes	No	Yes	Yes	1	0
No	No	Yes	No	Yes	No	Yes	No	1	0
No	No	Yes	No	Yes	No	No	Yes	1	0
No	No	Yes	No	Yes	No	No	No	1	0
No	No	Yes	No	No	Yes	Yes	Yes	1	0
No	No	Yes	No	No	Yes	Yes	No	1	0
No	No	Yes	No	No	Yes	No	Yes	0	1
No	No	Yes	No	No	Yes	No	No	0	1
No	No	Yes	No	No	No	Yes	Yes	1	0
No	No	Yes	No	No	No	Yes	No	1	0
No	No	Yes	No	No	No	No	Yes	0	1
No	No	Yes	No	No	No	No	No	0	1
No	No	No	Yes	Yes	Yes	Yes	Yes	1	0
No	No	No	Yes	Yes	Yes	Yes	No	1	0
No	No	No	Yes	Yes	Yes	No	Yes	1	0
No	No	No	Yes	Yes	Yes	No	No	1	0

								Principle 5	
Scope	Time	Cost	Quality	Communications	Resources	Safety	Risk	Yes	No
No	No	No	Yes	Yes	No	Yes	Yes	1	0
No	No	No	Yes	Yes	No	Yes	No	1	0
No	No	No	Yes	Yes	No	No	Yes	1	0
No	No	No	Yes	Yes	No	No	No	1	0
No	No	No	Yes	No	Yes	Yes	Yes	1	0
No	No	No	Yes	No	Yes	Yes	No	1	0
No	No	No	Yes	No	Yes	No	Yes	0	1
No	No	No	Yes	No	Yes	No	No	0	1
No	No	No	Yes	No	No	Yes	Yes	1	0
No	No	No	Yes	No	No	Yes	No	1	0
No	No	No	Yes	No	No	No	Yes	0	1
No	No	No	Yes	No	No	No	No	0	1
No	No	No	No	Yes	Yes	Yes	Yes	1	0
No	No	No	No	Yes	Yes	Yes	No	1	0
No	No	No	No	Yes	Yes	No	Yes	1	0
No	No	No	No	Yes	Yes	No	No	1	0
No	No	No	No	Yes	No	Yes	Yes	1	0
No	No	No	No	Yes	No	Yes	No	1	0
No	No	No	No	Yes	No	No	Yes	1	0
No	No	No	No	Yes	No	No	No	0	1
No	No	No	No	No	Yes	Yes	Yes	1	0
No	No	No	No	No	Yes	Yes	No	1	0
No	No	No	No	No	Yes	No	Yes	0	1
No	No	No	No	No	Yes	No	No	0	1
No	No	No	No	No	No	Yes	Yes	1	0
No	No	No	No	No	No	Yes	No	0	1
No	No	No	No	No	No	No	Yes	0	1
No	No	No	No	No	No	No	No	0	1

APPENDIX 4 – Tunnel Trains Simulation Information

A4.1 Model Overview

The tunnel train simulation model is a single agent example application where the agent is employed to determine when a second muck train should be employed in the simulation so to balance the spoil removal from the TBM. The agent employed in this simulation model makes use of a belief network for its intelligence. The model was constructed using the Common (Hajjar and AbouRizk, 2002), Functional Modeling (Mohammed, 2002), and CEM Tunnel 2 (Mohammed, 2002) simulation templates. The following section provides the information to construct the model.

A4.2 Model Construction

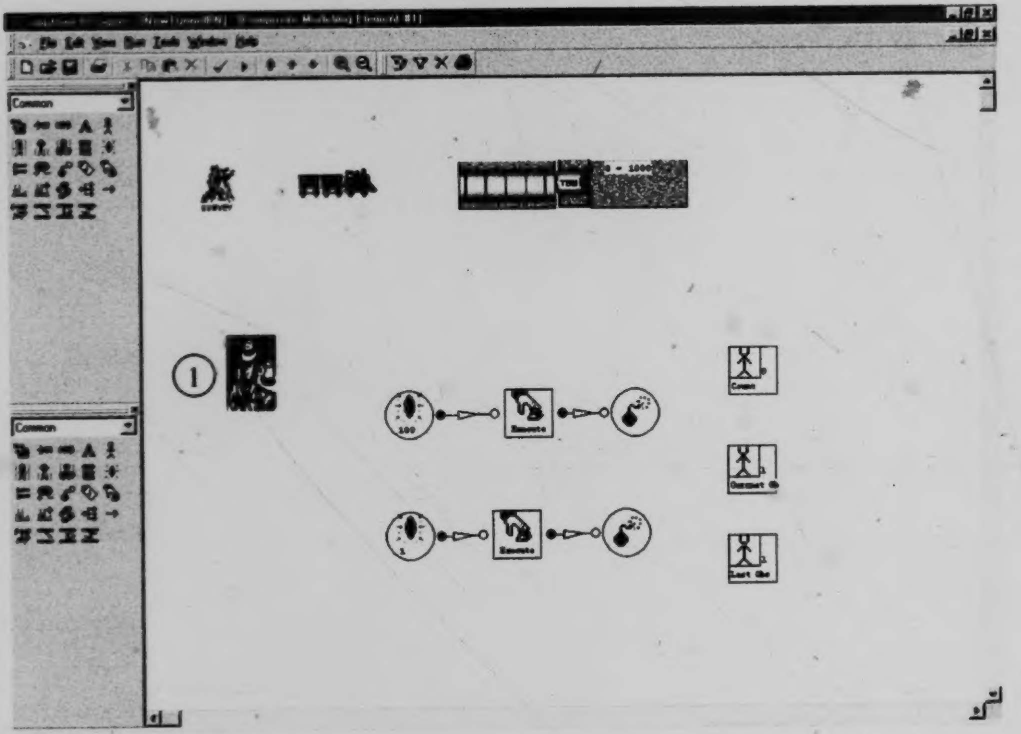

Common
Common
1
100
Execute
1
Execute
Count
Last Obs

Figure A4-1. Project Level View

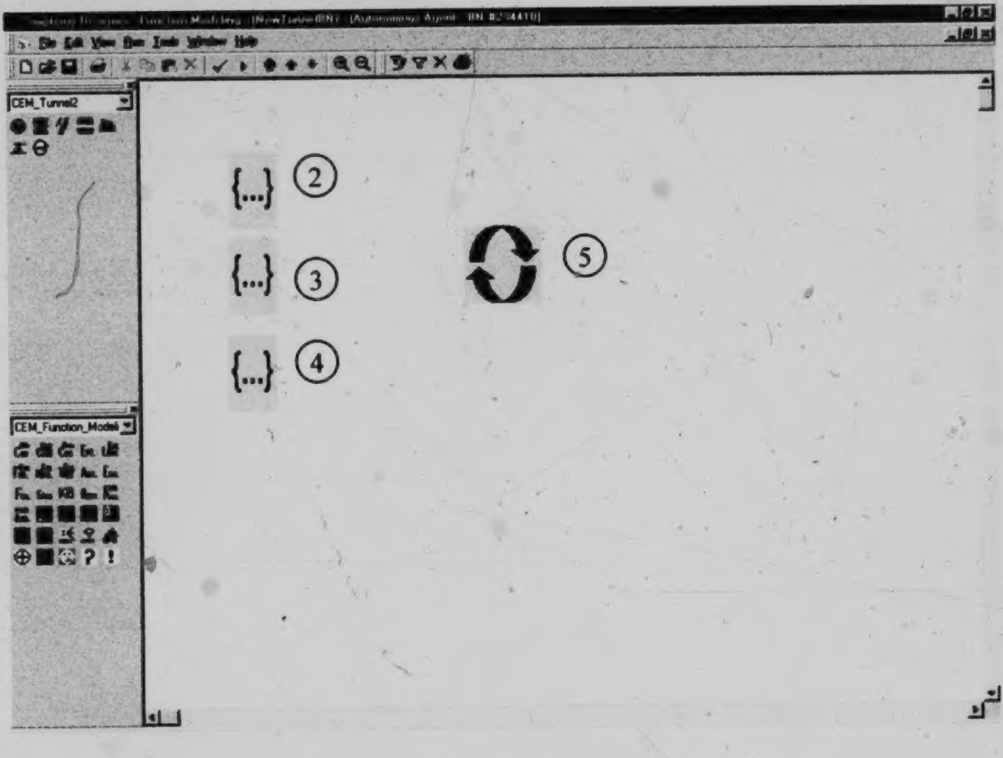
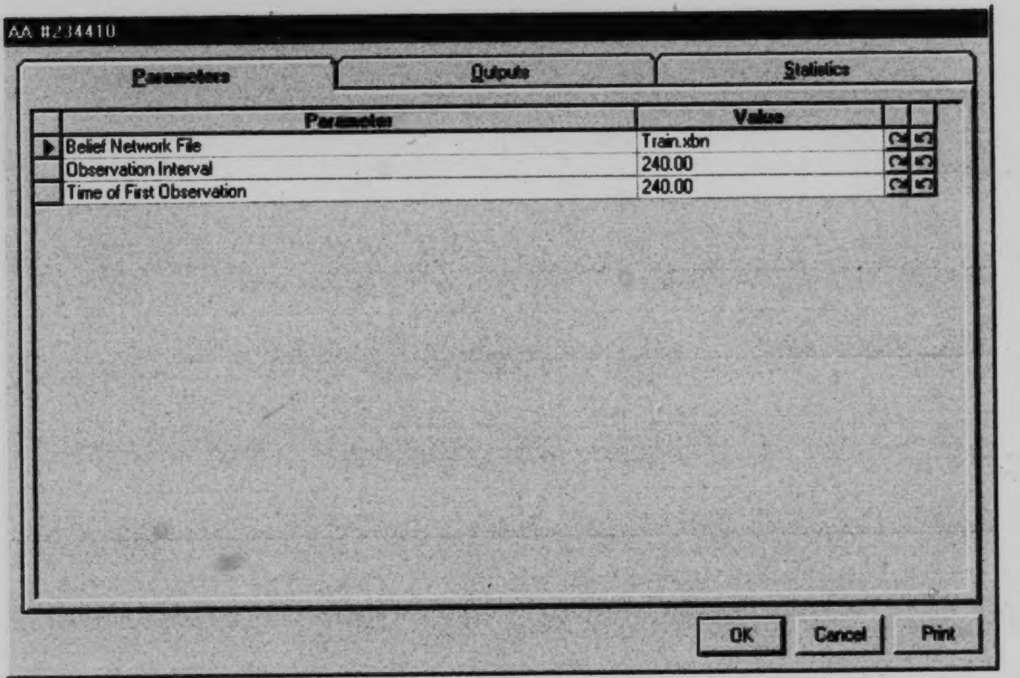

CEM_Tunnel2
CEM_Function_Models
②
③
④
⑤

Figure A4-2. Parent Agent Element 6 Child Window

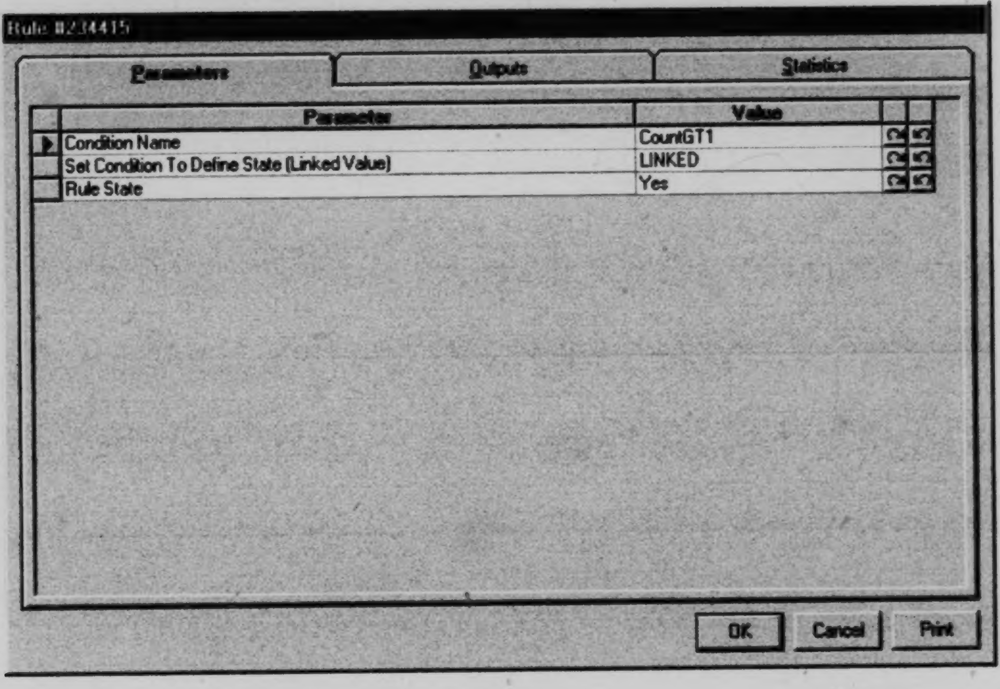
A4.2.1 Element 1 Parent Agent Element


AA #234410
Parameters
Outputs
Statistics
OK
Cancel
Print

Parameter	Value
Belief Network File	Train.xbn
Observation Interval	240.00
Time of First Observation	240.00

Figure A4-3. Parent Agent Modeling Element Parameters

A4.2.2 Element [illegible] Condition Element


Rule #234415
Parameters
Outputs
Statistics
OK
Cancel
Print

	Parameter	Value
▶	Condition Name	CountGT1
	Set Condition To Define State (Linked Value)	LINKED
	Rule State	Yes

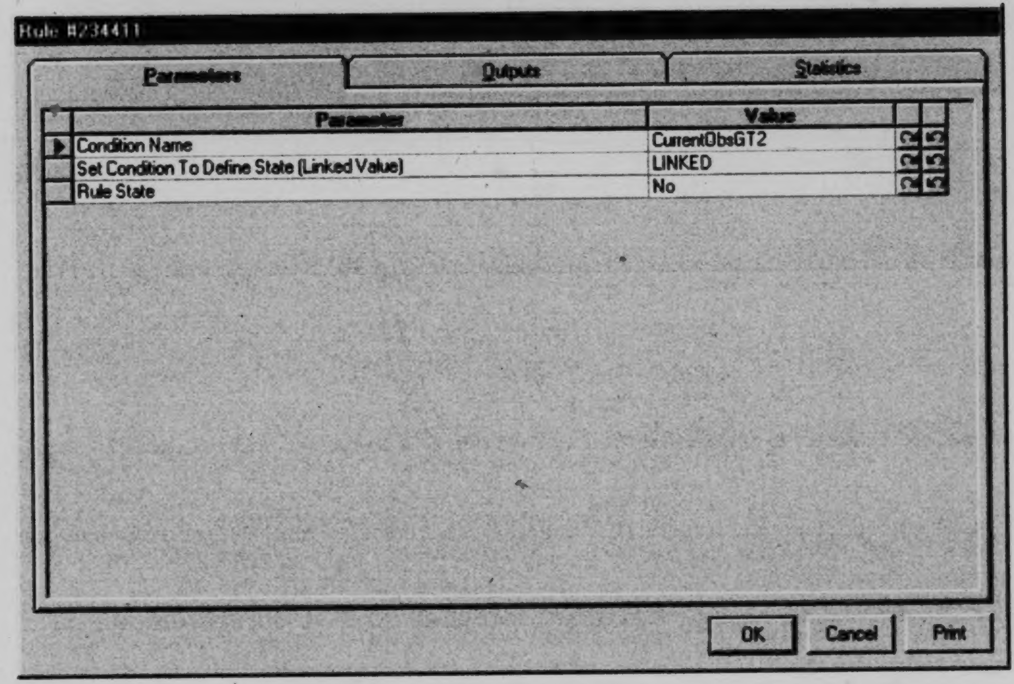
Figure A4-4. Condition Modeling Element Parameters

A4.2.2.1 Set Condition To Define State (Linked Value) Linked Code

```
Public Function F234416(ob As CFCSim_ModelingElementInstance)

    If Elements("13")("Current") >= 1 Then
        ob("RuleState") = "Yes"
    ElseIf Elements("13")("Current") < 1 Then
        ob("RuleState") = "No"
    Else
```

```
            ob("RuleState") = "Unobserved"

        End If


  F234416 = 0


End Function
```

A4.2.3 Element 3 Condition Element


Rule #234411
Parameters
Outputs
Statistics
OK
Cancel
Print

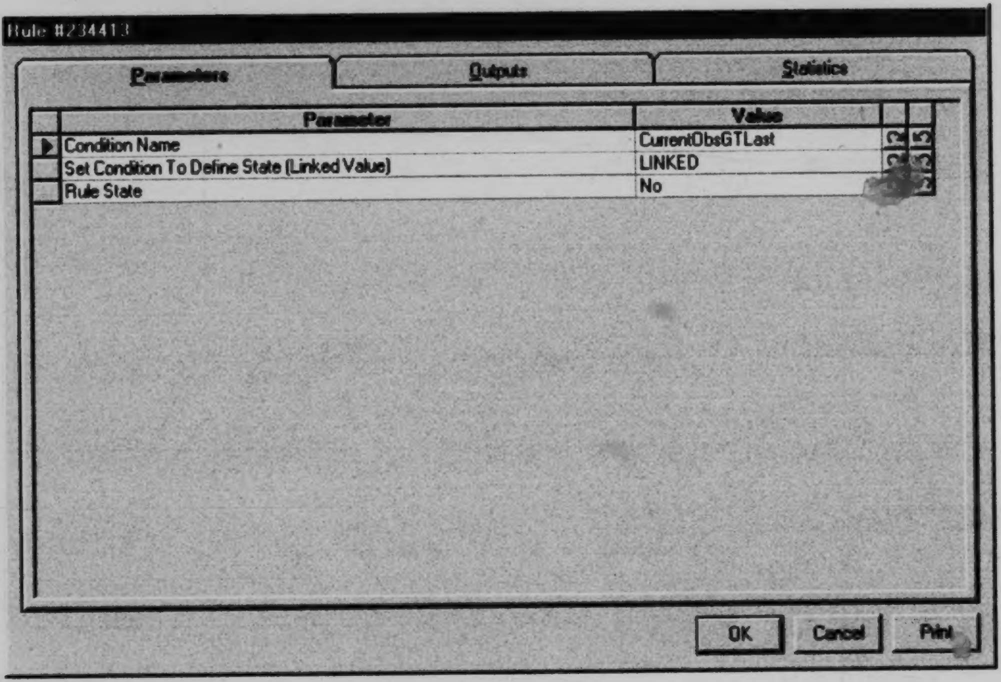
Parameter	Value
Condition Name	CurrentObsGT2
Set Condition To Define State (Linked Value)	LINKED
Rule State	No

Figure A4-5. Condition Modeling Element Parameters

A4.2.3.1 Set Condition To Define State (Linked Value) Linked Code

```
Public Function F234412(ob As CFCSim_ModelingElementInstance)

        If Elements("14")("Current") >= 2 Then
                ob("RuleState") = "Yes"
         ElseIf  Elements("14")("Current") < 2 Then
                ob("RuleState") = "No"
        Else
                ob("RuleState") = "Unobserved"
        End If

  F234412 = 0

End Function
```

A4.2.4 Element 4 Condition Element


Rule #234413
Parameters
Outputs
Statistics
OK
Cancel
Print

Parameter	Value
Condition Name	CurrentObsGTLast
Set Condition To Define State (Linked Value)	LINKED
Rule State	No

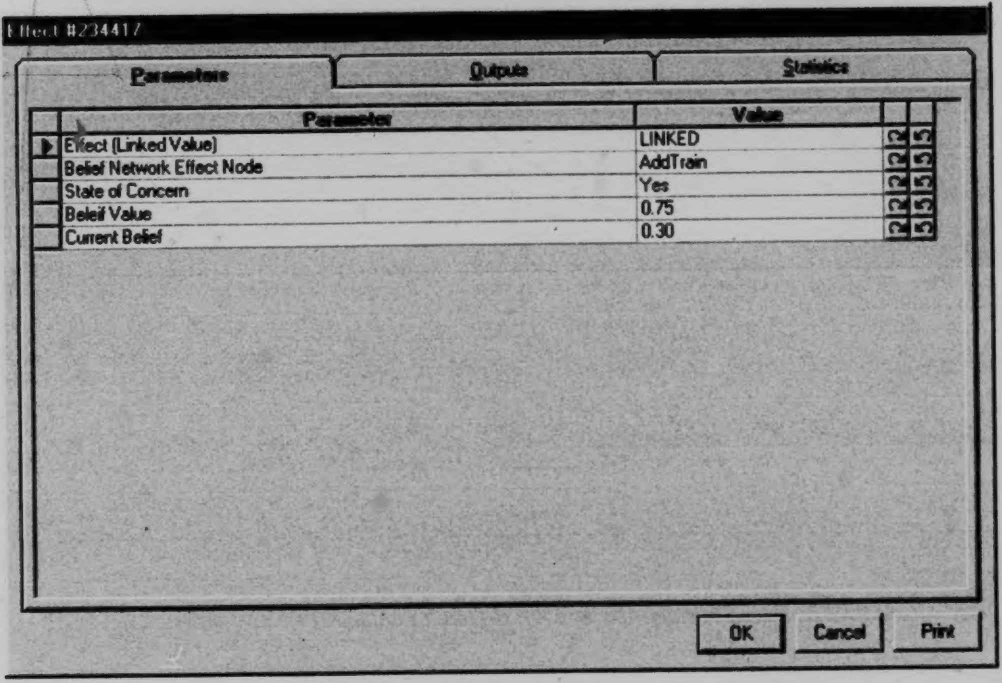
Figure A4-6. Condition Modeling Element Parameters

A4.2.4.1 Set Condition To Define State (Linked Value) Linked Code

```
Public Function F234414(ob As CFCSim_ModelingElementInstance)


    If Elements("14")("Current") >= Elements("15")("Current") Then

        ob("RuleState") = "Yes"

    ElseIf Elements("14")("Current") < Elements("15")("Current") Then

        ob("RuleState") = "No"

    Else
```

```
            ob("RuleState") = "Unobserved"

        End If


    F234414 = 0


End Function
```

A4.2.5 Element 5 Effect Element



Effect #234417
Parameters
Outputs
Statistics
OK
Cancel
Print

Parameter	Value
Effect (Linked Value)	LINKED
Belief Network Effect Node	AddTrain
State of Concern	Yes
Beleif Value	0.75
Current Belief	0.30

Figure A4-7. Effect Modeling Element Parameters

A4.2.5.1 Effect (Linked Value)Linked Code

```
Public Function F234418(ob As CFCSim_ModelingElementInstance)
  F234418 =  "~Element Group~"
   Dim ob1 As CFCSim_ModelingElementInstance
   Dim kb As CFCSim_ModelingElementInstance
   Dim Ent As CFCSim_Entity
       Set Ent=ob.AddEntity
       Ent("Function")="Nothing"
   Set ob1 = ob.Parent.Parent.AddElement("CEM_Tunnel2_Train",71.0,169.0)
       ob1("Location")="Undercut"
       ob1("Load")="Empty"
   Elements("235")("ElementGroup").Collection.Add(ob1.Id,ob1.Id)
   For Each kb In ob1.ChildElements
         If kb.ElementType="CEM_FM_TransitionKB" Then
               kb.OnSimulationTransferIn Ent,Nothing,Nothing
                 Exit For
               End If
        Next
 End Function
```

APPENDIX 5 – Truck Queuing Simulation Information

A5.1 Model Overview

The truck queuing simulation model is an example of a single agent application employed within an earthmoving simulation environment to determine the best allocation of haul trucks for a single excavator based on queuing characteristics occurring in the model. The model was constructed using the hybrid Cyclone Simulation Template developed by Van Tol (2000) and the Common Simulation Template (Hajjar and AbouRizk, 2002). The simulation agent makes use of the belief network "Play" for its reasoning. The details regarding this belief network can be found in Appendix 3. The following section provides the necessary information to reconstruct the model as it was for this research.

A5.2 Model Construction

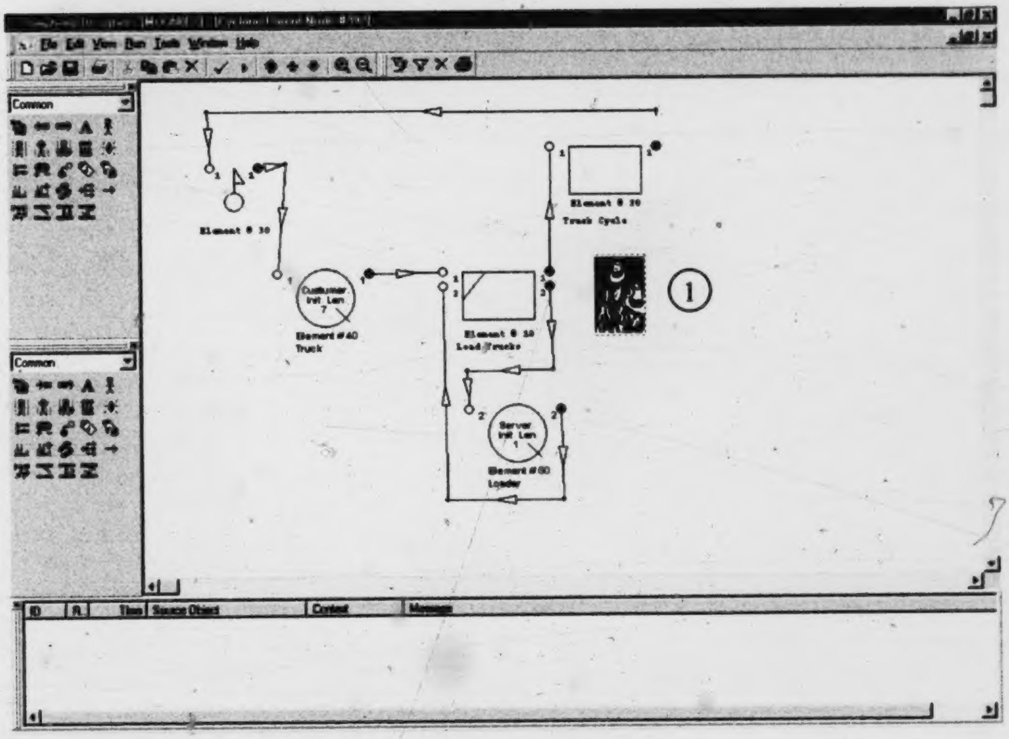

File Edit View Run Tools Window Help
Common
Common
Element # 30
Customer Init Len 7
Element #40
Truck
Element # 10
Load Trucks
Element # 30
Truck Cycle
Server Init Len 1
Element #60
Loader
①
ID
Time
Source Object
Context
Message

Figure A5-1. Cyclone Parent Element Child Window

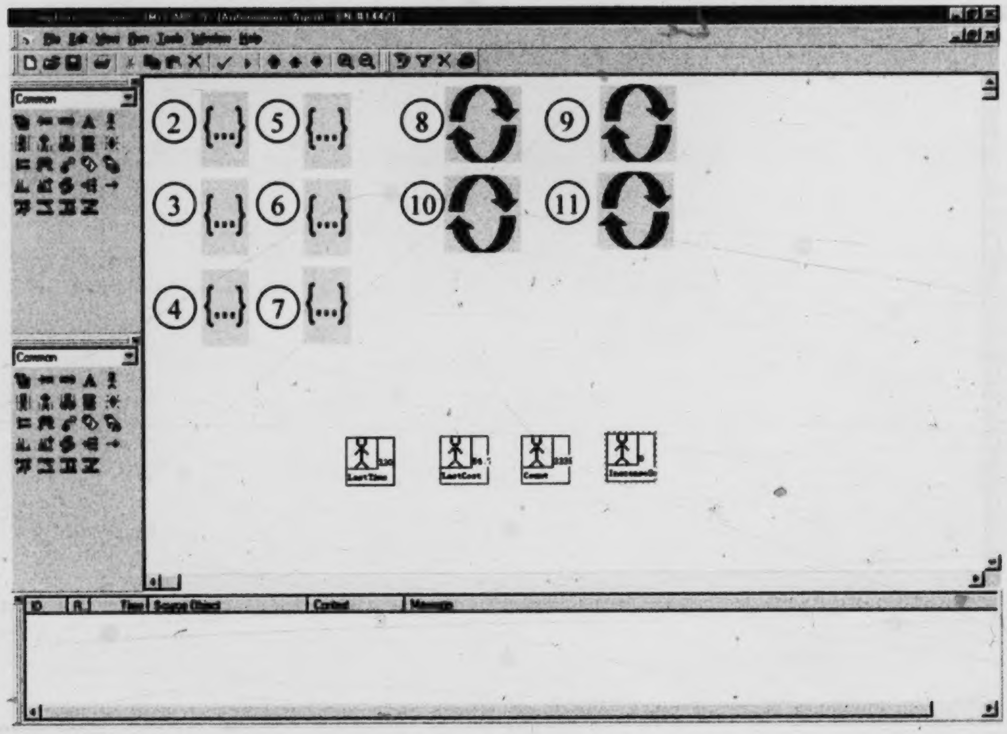
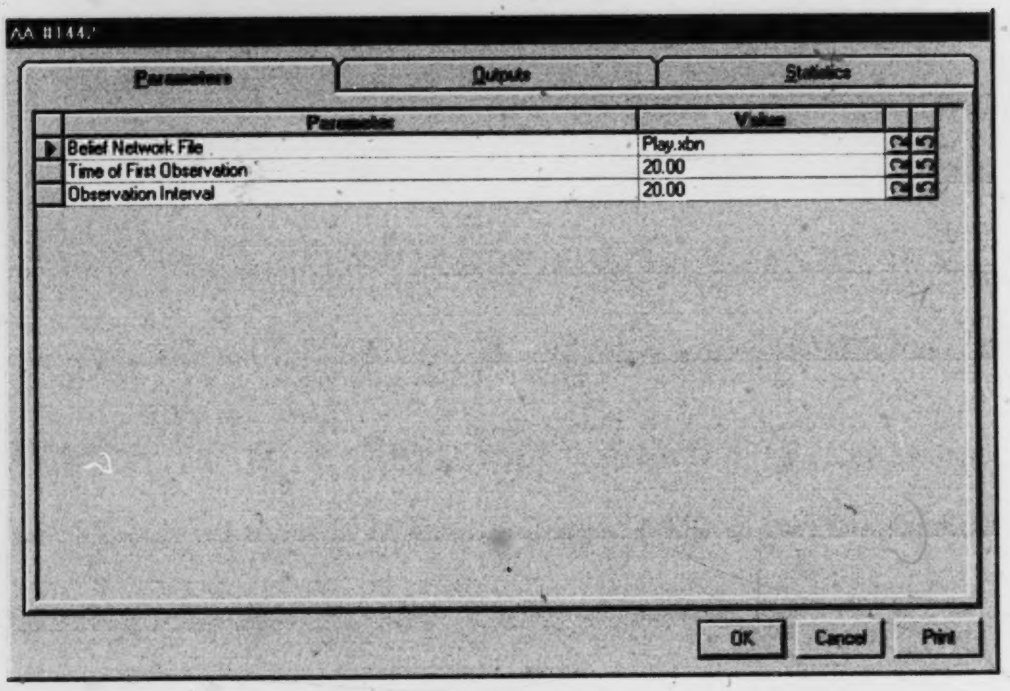

File
Edit
View
Run
Tools
Window
Help
Common
Common
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
{...}
LastTime
LastCost
Count
ID
R
Time
Source Object
Context
Message

Figure A5-2. Agent Parent Element Child Window

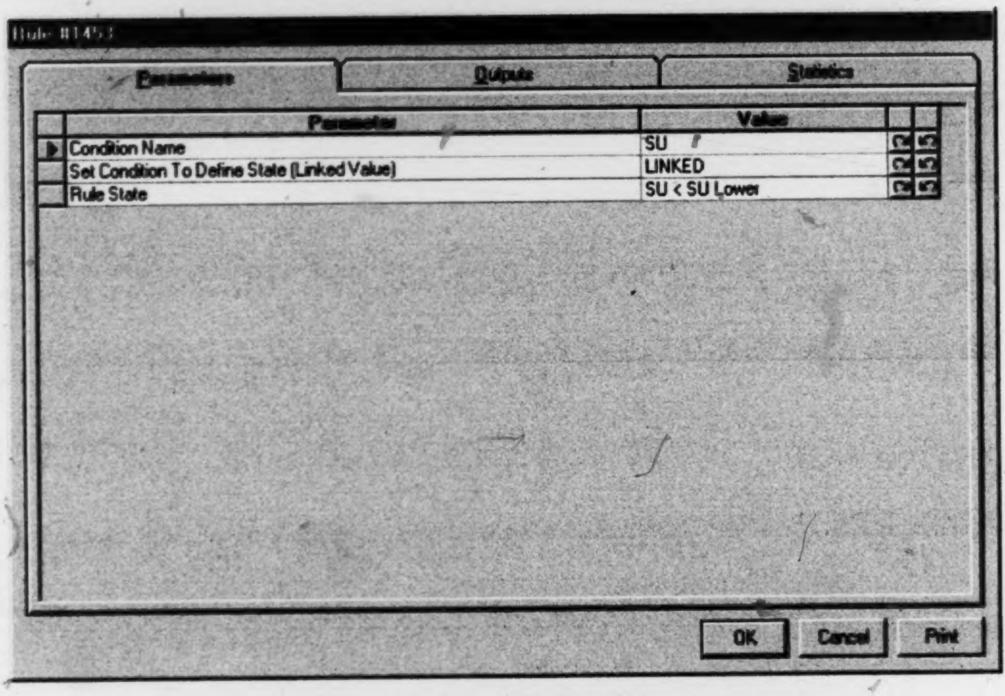
A5.2.1 Element 1 Agent Element


AA #144
Parameters
Outputs
Statistics
OK
Cancel
Print

Parameter	Value
Belief Network File	Play.xbn
Time of First Observation	20.00
Observation Interval	20.00

Figure A5-3. Agent Modeling Element Parameters

A5.2.2 Element 2 Server Utilization Condition Element


Rule #1453
Parameters
Outputs
Statistics
OK
Cancel
Print

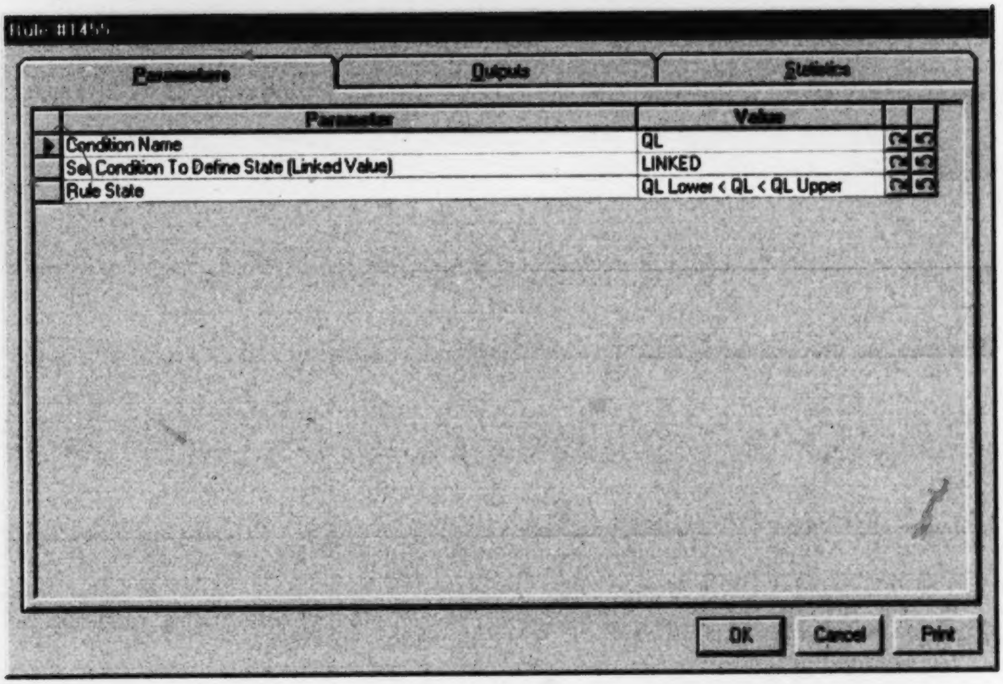
Parameter	Value
Condition Name	SU
Set Condition To Define State (Linked Value)	LINKED
Rule State	SU < SU Lower

Figure A5-4. Server Utilization Condition Modeling Element Parameters

A5.2.2.1 Set Condition To Define State (Linked Value) Linked Code

```
Public Function F1454(ob As CFCSim_ModelingElementInstance)
        Dim Lower As Single
        Dim Upper As Single

        Lower = .25
        Upper =.95
        'below lower tolerance
        If Elements("10").stat("ServerUtilization").GlobalAverage < Lower Then
                ob("RuleState") = "SU < SU Lower"
        'above upper tolerance
        ElseIf Elements("10").stat("ServerUtilization").GlobalAverage > Upper Then
                ob("RuleState") = "SU > SU Upper"
        ElseIf  Elements("10").stat("ServerUtilization").GlobalAverage  >=  Lower  And
Elements("10").stat("ServerUtilization").GlobalAverage <= Upper  Then
                ob("RuleState") = "SU Lower < SU < SU Upper"
        Else
                ob("RuleState") = "Unobserved"
        End If

    F1454 = 0
End Function
```

A5.2.3 Element 3 Average Queue Length Condition Element


Rule #1455
Parameters
Outputs
Statistics
OK
Cancel
Print

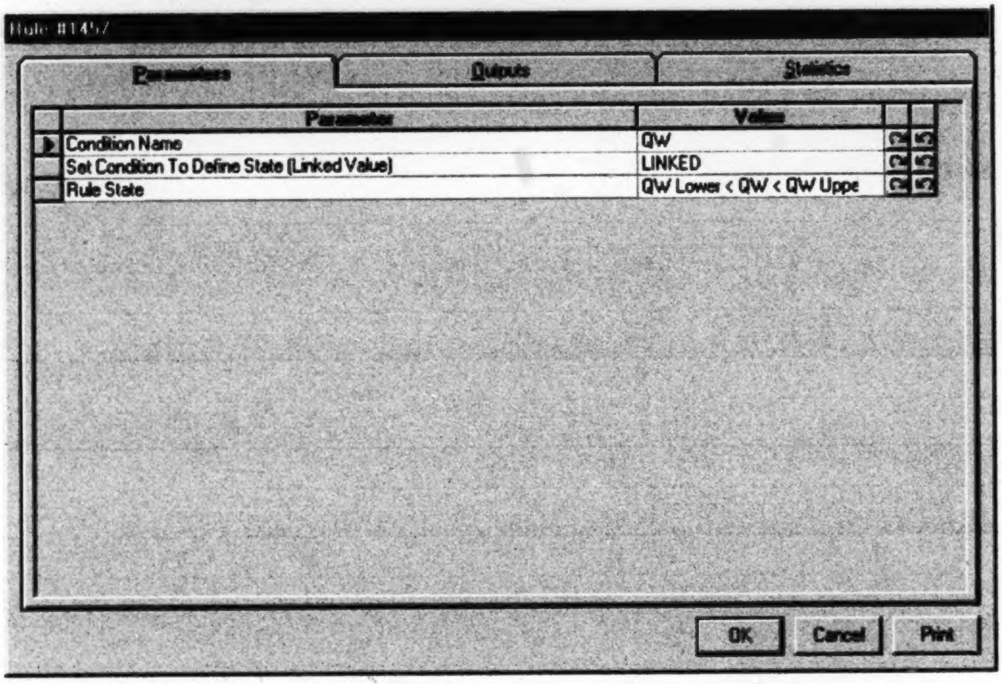
Parameter	Value
Condition Name	QL
Set Condition To Define State (Linked Value)	LINKED
Rule State	QL Lower < QL < QL Upper

Figure A5-5. Average Queue Length Condition Modeling Element Parameters

A5.2.3.1 Set Condition To Define State (Linked Value) Linked Code

```
Public Function F1456(ob As CFCSim_ModelingElementInstance)

        Dim Lower As Single

        Dim Upper As Single


        Lower = 0

        Upper =1

        'below lower tolerance
        If Elements("7").stat("File_Queue_FileLength").GlobalAverage < Lower Then
                ob("RuleState") = "QL < QL Lower"
        'above upper tolerance
        ElseIf Elements("7").stat("File_Queue_FileLength").GlobalAverage > Upper Then
                ob("RuleState") = "QL > QL Upper"
        ElseIf Elements("7").stat("File_Queue_FileLength").GlobalAverage >= Elements("7").stat("File_Queue_FileLength").GlobalAverage <= Upper  Then
                ob("RuleState") = "QL Lower < QL < QL Upper"
        Else
                ob("RuleState") = "Unobserved"
        End If

  F1456 = 0
End Function
```

A5.2.4 Element 4 Average Queue Wait Condition Element


Rule #1457
Parameters
Outputs
Statistics
OK
Cancel
Print

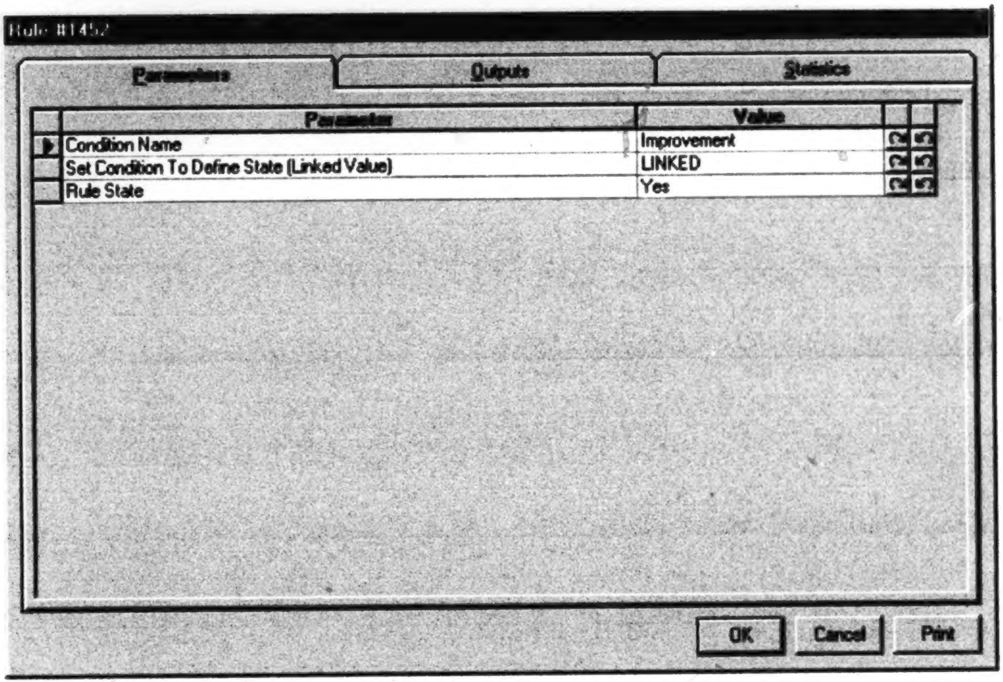
Parameter	Value
Condition Name	QW
Set Condition To Define State (Linked Value)	LINKED
Rule State	QW Lower < QW < QW Uppe

Figure A5-6. Average Queue Wait Condition Modeling Element Parameters

A5.2.4.1 Set Condition To Define State (Linked Value) Linked Code

```
Public Function F1458(ob As CFCSim_ModelingElementInstance)

    Dim Lower As Single

    Dim Upper As Single


    Lower = 0

    Upper = 5

    'below lower tolerance
    If Elements("7").stat("File_Queue_WaitingTime").GlobalAverage < Lower Then
            ob("RuleState") = "QW < QW Lower"
    'above upper tolerance
    ElseIf Elements("7").stat("File_Queue_WaitingTime").GlobalAverage > Upper Then
            ob("RuleState") = "QW > QW Upper"
    ElseIf          Elements("7").stat("File_Queue_WaitingTime").GlobalAverage      >=
Elements("7").stat("File_Queue_WaitingTime").GlobalAverage <= Upper  Then
            ob("RuleState") = "QW Lower < QW < QW Upper"
    Else
            ob("RuleState") = "Unobserved"
    End If

  F1458 = 0
End Function
```

A5.2.5 Element 5 Improvement Condition Element


Rule #1452
Parameters
Outputs
Statistics
OK
Cancel
Print

Parameter	Value
Condition Name	Improvement
Set Condition To Define State (Linked Value)	LINKED
Rule State	Yes

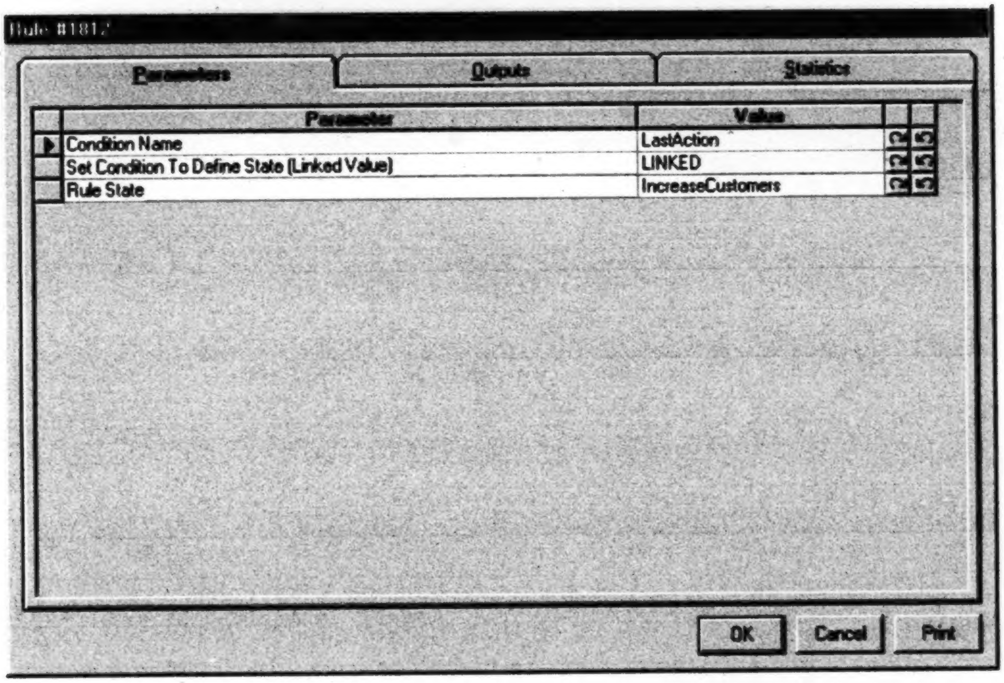
Figure A5-7. Improvement Condition Modeling Element Parameters

A5.2.5.1 Set Condition To Define State (Linked Value) Linked Code

```
Public Function F1463(ob As CFCSim_ModelingElementInstance)

Dim TCost As Double


TCost                                                                  =

Elements("10")("HrCost")*Elements("10")("NumInQueue")+Elements("7")("HrCost")*Elem

ents("7")("NumInQueue")
```

```
TCost        =(TCost        *(SimTime-Elements("21")("Total")))/(Elements("5")("NumEnt")-
Elements("23")("Total"))


Elements("21")("Total") = SimTime


If TCost < Elements("22")("Total") Then
        ob("RuleState") = "Yes"
Else
        ob("RuleState") = "No"
End If
Elements("22")("Total") = TCost
Elements("23")("Total") =Elements("5")("NumEnt")
F1463 = 0
End Function
```

A5.2.6 Element 6 Last Action Condition Element


Rule #1812
Parameters
Outputs
Statistics
OK
Cancel
Print

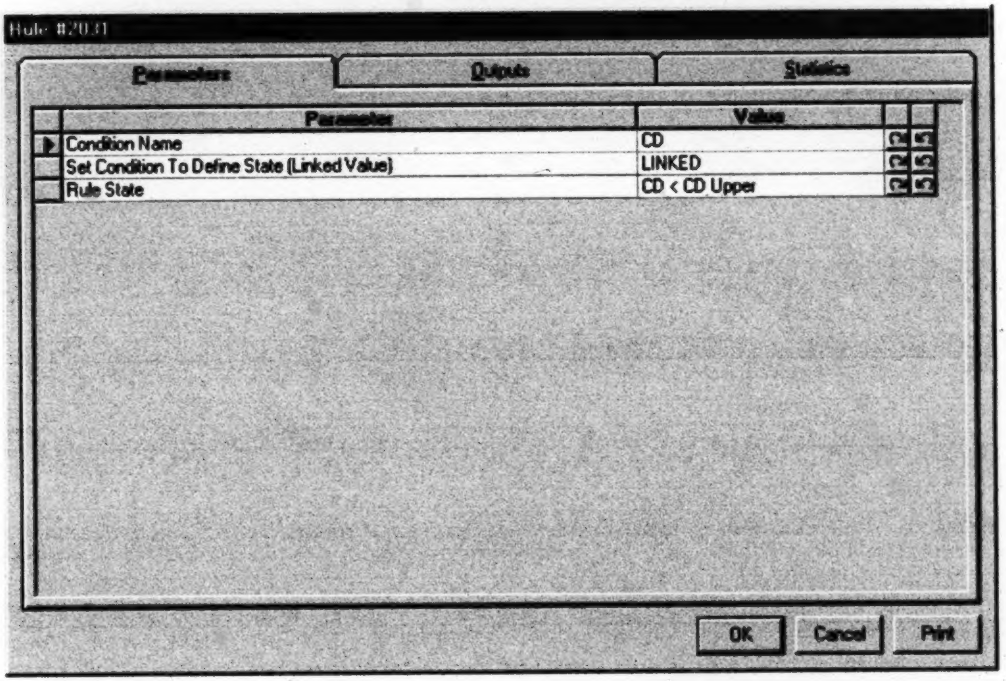
Parameter	Value
Condition Name	LastAction
Set Condition To Define State (Linked Value)	LINKED
Rule State	IncreaseCustomers

Figure A5-8. Last Action Condition Modeling Element Parameters

A5.2.6.1 Set Condition To Define State (Linked Value) Linked Code

```
Public Function F1813(ob As CFCSim_ModelingElementInstance)

ob("RuleState")= Elements("24")("ResName")

  F1813 = 0

End Function
```

A5.2.7 Element 7 Customer Delay Condition Element



Parameter	Value
Condition Name	CD
Set Condition To Define State (Linked Value)	LINKED
Rule State	CD < CD Upper

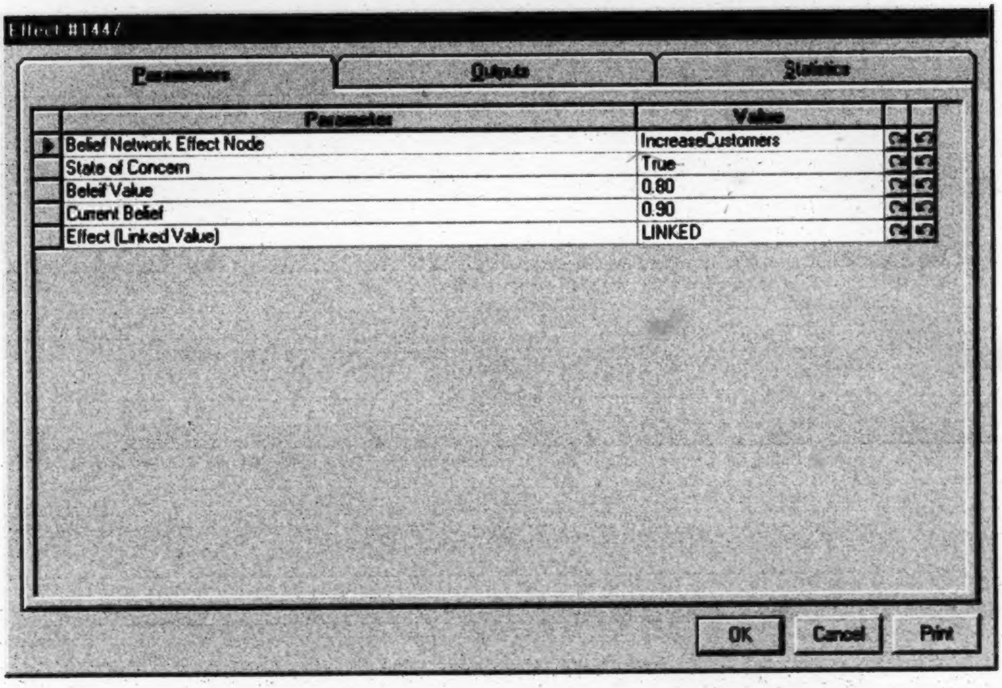
Figure A5-9. Customer Delay Condition Modeling Element Parameters

A5.2.7.1 Set Condition To Define State (Linked Value) Linked Code

```
Public Function F2186(ob As CFCSim_ModelingElementInstance)

    Dim Upper As Single

    Upper =.3

    If Elements("7").stat("CustomerDelayStat").GlobalAverage <= Upper Then

        ob("RuleState") = "CD < CD Upper"

    ElseIf Elements("7").stat("CustomerDelayStat").GlobalAverage > Upper Then
```

```
            ob("RuleState") = "CD > CD Upper"

        Else

            ob("RuleState") = "Unobserved"

        End If

    F2186 = 0

End Function
```

A5.2.8 Element 8 Increase Customer Effect Element



Effect #1447
Parameters
Outputs
Statistics
OK
Cancel
Print

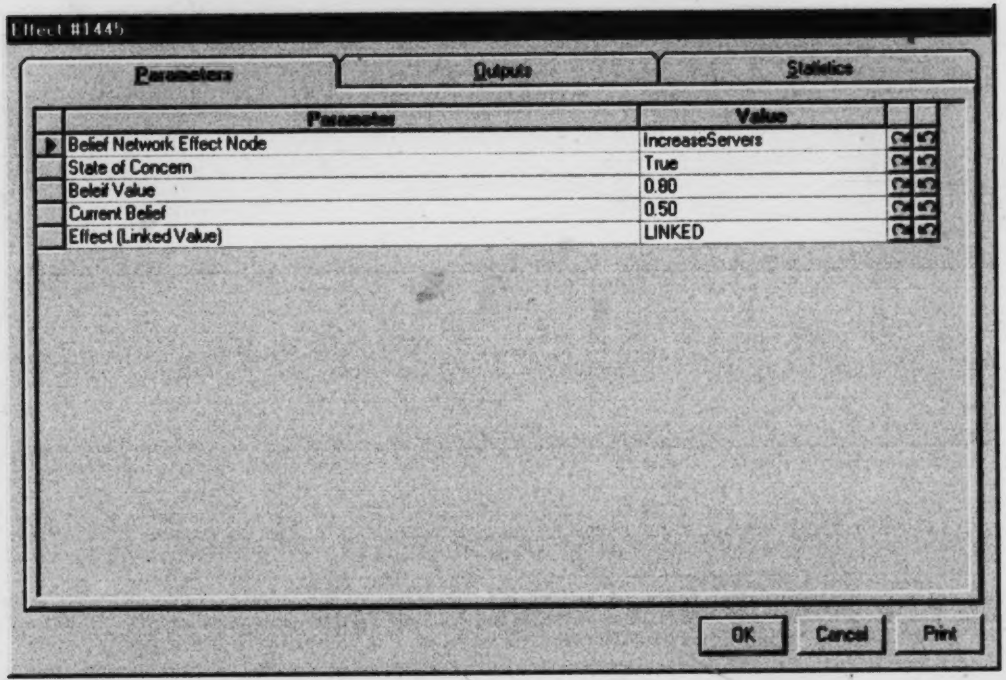
Parameter	Value
Belief Network Effect Node	IncreaseCustomers
State of Concern	True
Beleif Value	0.80
Current Belief	0.90
Effect (Linked Value)	LINKED

Figure A5-10. Increase Customer Effect Modeling Element Parameters

A5.2.8.1 Effect (Linked Value) Linked Code

```
Public Function F1448(ob As CFCSim_ModelingElementInstance)

If SimTime <> 0 Then

        If Elements("7")("NumInQueue")<10 Then

                Dim Child As CFCSim_ModelingElementInstance

                Dim Ent As CFCSim_Entity

                For Each Child In ob.Parent.Parent.ChildElements

                        If Child.ElementType= "DC_Customer_Queue" Then

                                Child("NumInQueue") = Child("NumInQueue") +1

                                Elements("24")("ResName") =  "IncreaseCustomers"

                                Set Ent = Child.AddEntity

                                        Ent.Attr("EKind") = "Customer" & Child("NodeID")

                                        Ent.Attr("Kind") = "QueueEnt"

                                        Ent.Attr("QEntType") = "Customer"

                                        Ent.Attr("FirstTime") = "Yes"

                                        Ent.Attr("StartDelay") = SimTime

                                        Ent.Attr("TotalDelay") = 0

                                        Ent.Attr("CycleStart") = SimTime

                                        Ent.Attr("TimeStamp1") = 0
                Child.File("Queue").Add Ent
            End If
        Next Child
    End If
End If
F1448 = 0
End Function
```

A5.2.9 Element 9 Increase Server Effect Element


Effect #1445
Parameters
Outputs
Statistics
OK
Cancel
Print

Parameter	Value
Belief Network Effect Node	IncreaseServers
State of Concern	True
Beleif Value	0.80
Current Belief	0.50
Effect (Linked Value)	LINKED

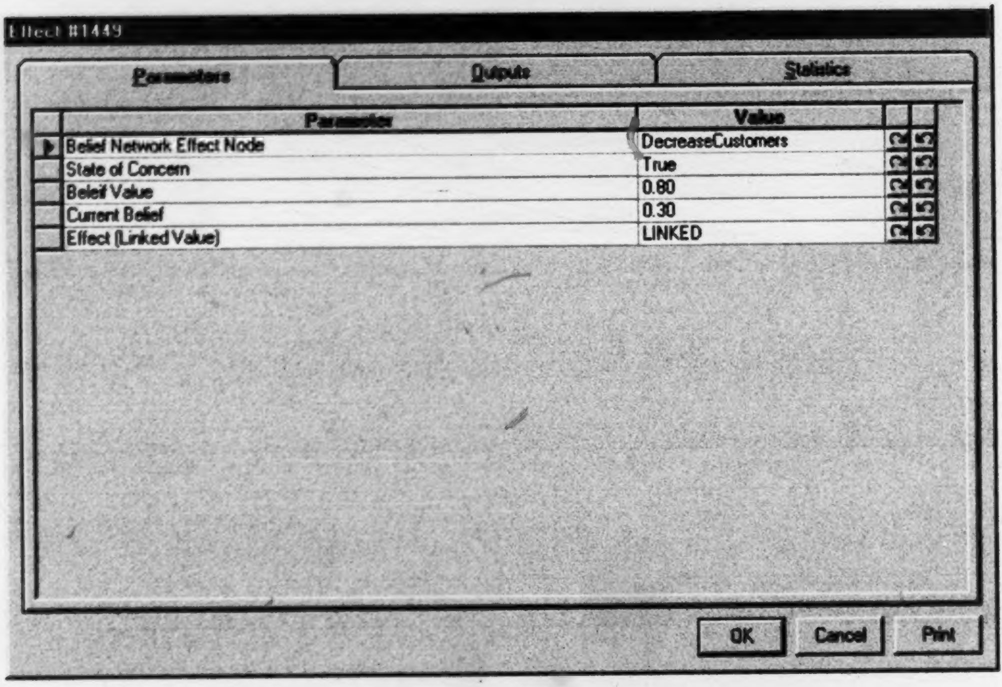
Figure A5-11. Increase Server Effect Modeling Element Parameters

A5.2.9.1 Effect (Linked Value) Linked Code

```
Public Function F1446(ob As CFCSim_ModelingElementInstance)

If SimTime <>0 Then

        If Elements("10")("NumInQueue") < 1 Then

                Dim Child As CFCSim_ModelingElementInstance

                Dim ent As CFCSim_Entity

                For Each Child In ob.Parent.Parent.ChildElements

                        If Child.ElementType = "DC_Server_Queue" Then

                                Elements("24")("ResName") =  "IncreaseServers"

                                Child("NumInQueue") = Child("NumInQueue") + 1

                                Set ent = Child.AddEntity

                                        ent.Attr("EKind") = "Server" & Child("NodeID")

                                        ent.Attr("Kind") = "QueueEnt"

                                        ent.Attr("QEntType") = "Server"

                                        ent.Attr("FirstTime") = "Yes"

                                        ent.Attr("StartDelay") = SimTime

                                        ent.Attr("TotalDelay") = 0

                                        ent.Attr("CycleStart") = SimTime

                                        ent.Attr("TimeStamp1") = 0
```

```
                    Child.File ("Queue").Add ent
                    End If
                Next Child
            End If
        End If
    F1446 = 0
    End Function
```

A5.2.10 Element 10 Decrease Customer Effect Element


Effect #1449
Parameters
Outputs
Statistics
OK
Cancel
Print

Parameter	Value
Belief Network Effect Node	DecreaseCustomers
State of Concern	True
Beleif Value	0.80
Current Belief	0.30
Effect (Linked Value)	LINKED

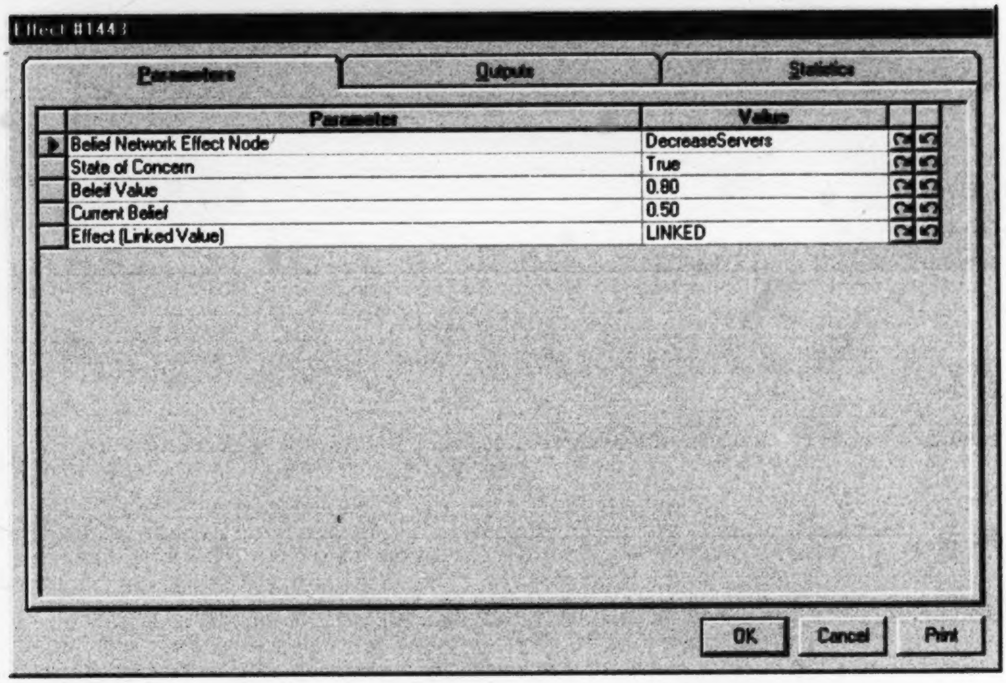
Figure A5-12. Decrease Customer Effect Modeling Element Parameters

A5.2.10.1 Effect (Linked Value) Linked Code

```
Public Function F1450(ob As CFCSim_ModelingElementInstance)

If SimTime <>0 Then

          If Elements("7")("NumInQueue")>2 Then

                    Dim Child As CFCSim_ModelingElementInstance

                    Dim ent As CFCSim_Entity

                    For Each Child In ob.Parent.Parent.ChildElements

                              If Child.ElementType ="DC_Customer_Queue" Then

                                        If Child.File("Queue").Length>0 Then

                                                  Elements("24")("ResName") =  "DecreaseCustomers"

                                                  Child("NumInQueue") = Child("NumInQueue") - 1

                                                  Set ent = Child.File("Queue").Pop

                                                  ob.DeleteEntity ent

                                        End If

                              End If

                    Next Child

          End If

End If

F1450 = 0
```

```
End Function
```

A5.2.11 Element 11 Decrease Server Effect Element


Effect #1443
Parameters
Outputs
Statistics
OK
Cancel
Print

Parameter	Value
Belief Network Effect Node	DecreaseServers
State of Concern	True
Beleif Value	0.80
Current Belief	0.50
Effect (Linked Value)	LINKED

Figure A5-13. Decrease Server Effect Modeling Element Parameters

A5.2.11.1 Effect (Linked Value) Linked Code

```
Public Function F1444(ob As CFCSim_ModelingElementInstance)

If SimTime <>0 Then

        If Elements("10")("NumInQueue") > 1 Then

                Dim Child As CFCSim_ModelingElementInstance
```

```
                    Dim Ent As CFCSim_Entity

                    For Each Child In ob.Parent.Parent.ChildElements

                        If Child.ElementType ="DC_Server_Queue" Then

                            If Child.File("Queue").Length>0 Then

                                Elements("24")("ResName") =  "DecreaseServers"

                                Child("NumInQueue") = Child("NumInQueue") - 1

                                Set Ent = Child.File("Queue").Pop

                                ob.DeleteEntity Ent

                            End If

                        End If

                    Next Child

                End If

            End If

            F1444 = 0

            End Function
```

APPENDIX 6 – Bridge Abutment Construction Simulation Information

A6.1 Model Overview

The bridge abutment construction simulation model is a single agent example application where the agent is employed to interpret weather data from an information source and apply the effects of weather through a productivity factor to the activities found in the simulation environment. This example was taken from Wales (1994). The model was constructed using the Common Simulation Template (Hajjar and AbouRizk, 2002) with the exception of the Task Modeling Element that was modified for this example. The agent in the model makes use of a "weather" neural network developed by Wales (1994). Information regarding this neural network can be found in Appendix 2. The following section provides the model construction information.

A6.2 Model Construction

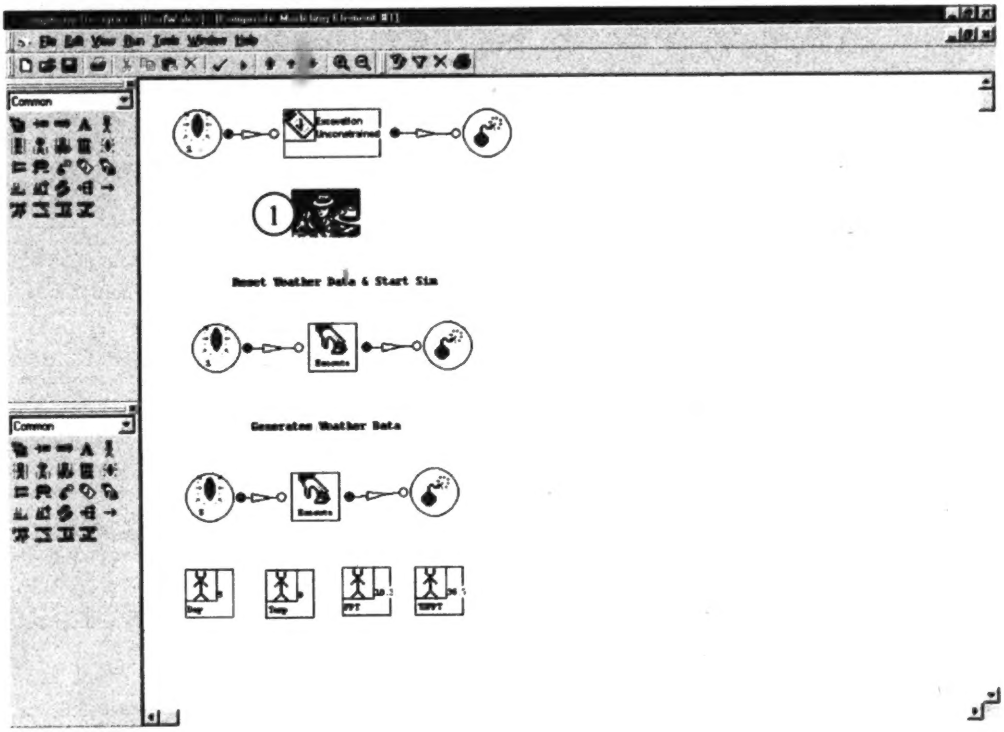

File
Edit
View
Run
Tools
Window
Help
Common
Execution
Unconstrained
1
Reset Weather Data & Start Sim
Generate Weather Data
Common
Day
Temp
PPT
TPPT

Figure A6-1. Project Level View

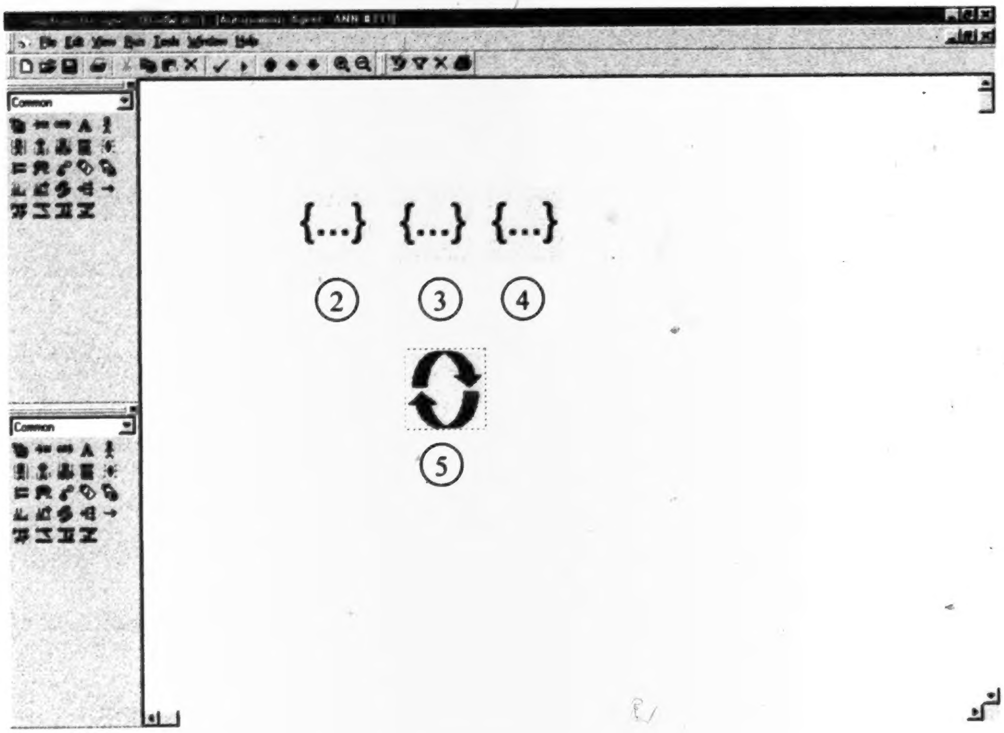
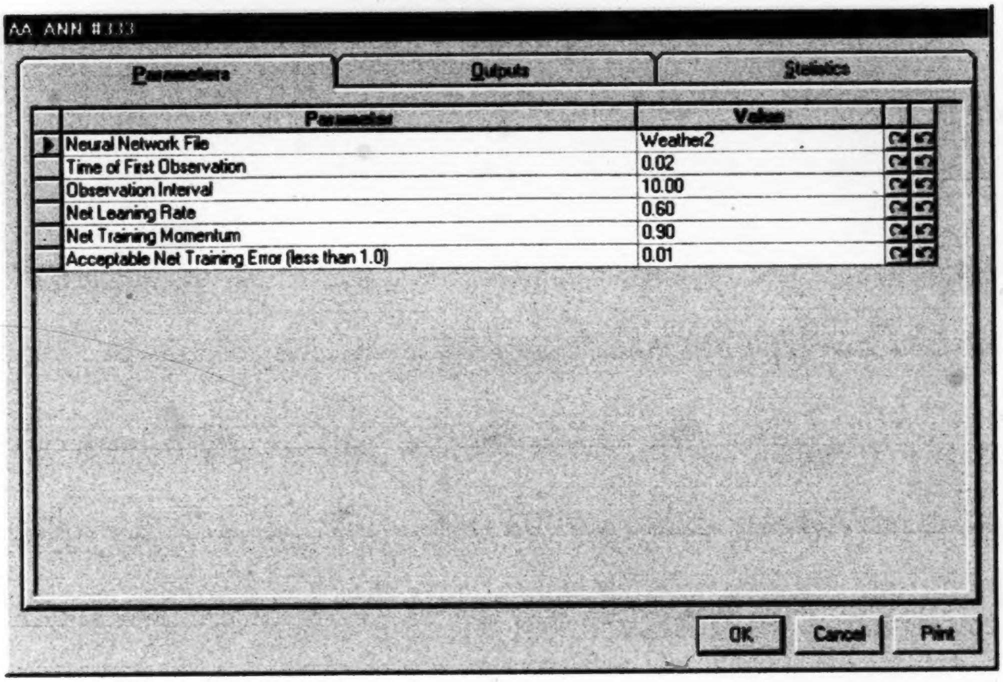

Common
Common
{...}
{...}
{...}
②
③
④
⑤

Figure A6-2. Agent Parent Element Child Window

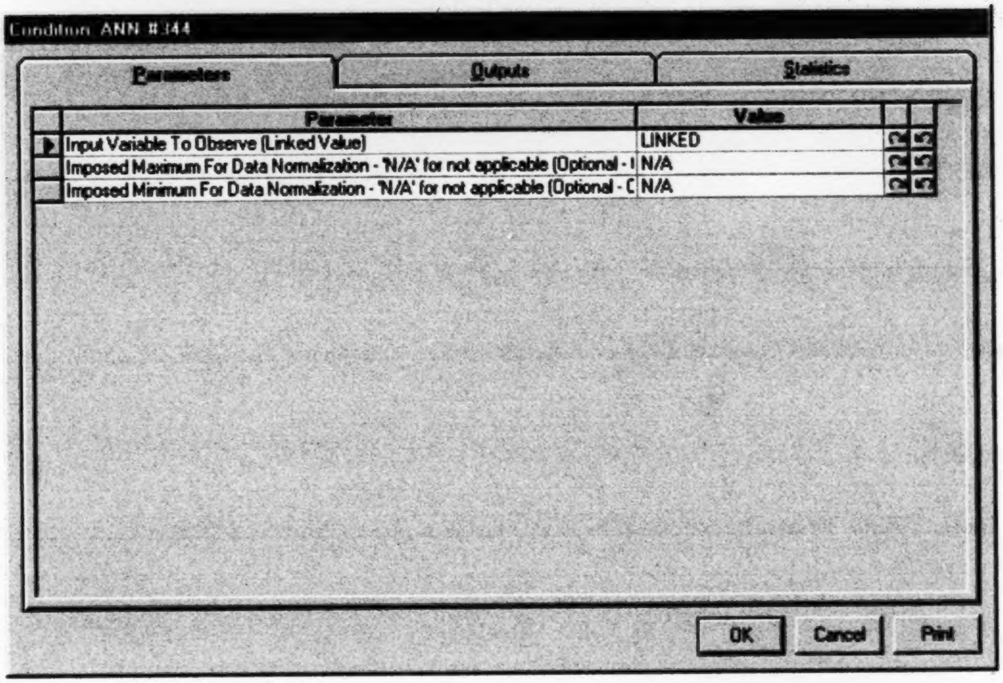
A6.2.1 Element 1 Agent Element


AA ANN #333
Parameters
Outputs
Statistics
OK
Cancel
Print

Parameter	Value
Neural Network File	Weather2
Time of First Observation	0.02
Observation Interval	10.00
Net Leaning Rate	0.60
Net Training Momentum	0.90
Acceptable Net Training Error (less than 1.0)	0.01

Figure A6-3. Agent Modeling Element Parameters

A6.2.2 Element 2 Average Daily Temperature Condition Element


Condition: ANN #344
Parameters
Outputs
Statistics
OK
Cancel
Print

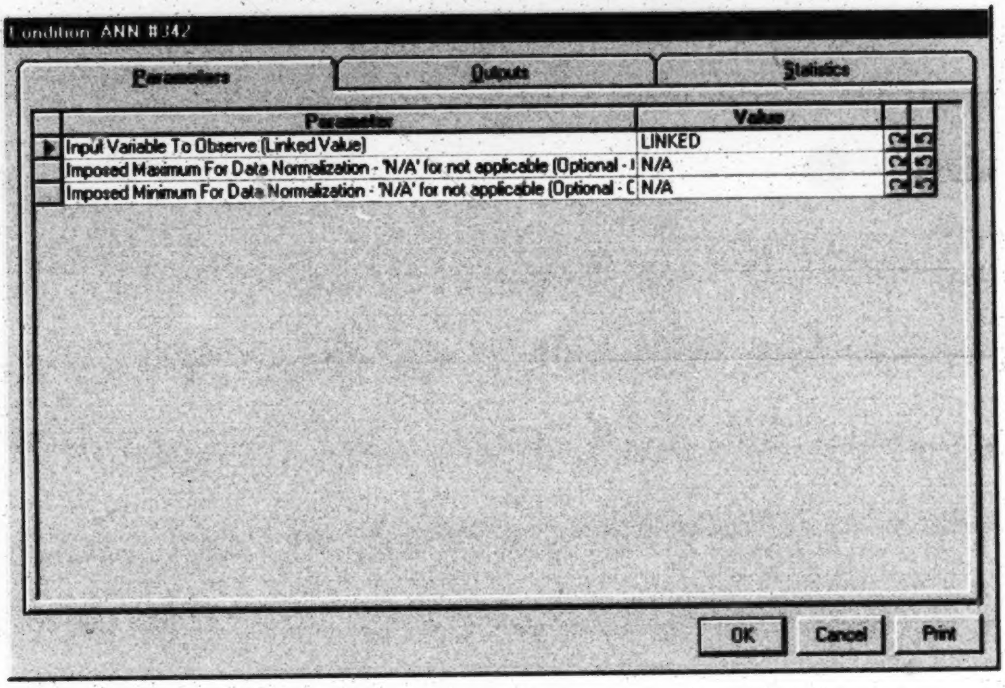
Parameter	Value
Input Variable To Observe (Linked Value)	LINKED
Imposed Maximum For Data Normalization - 'N/A' for not applicable (Optional - I	N/A
Imposed Minimum For Data Normalization - 'N/A' for not applicable (Optional - C	N/A

Figure A6-4. Average Daily Temperature Condition Element Parameters

A6.2.2.1 Input Variable To Observe (Linked Value) Linked Code

```
Public Function F345(ob As CFCSim_ModelingElementInstance)

   F345 = Elements("12")("Total")

End Function
```

A6.2.3 Element 3 Precipitation Condition Element


Condition ANN #342
Parameters
Outputs
Statistics
OK
Cancel
Print

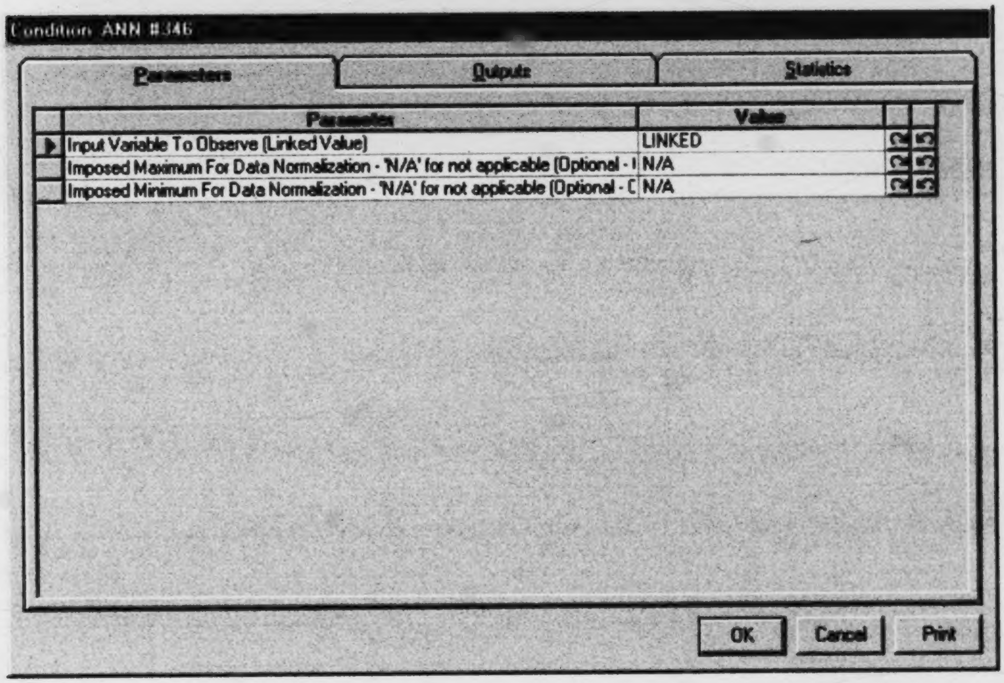
	Parameter	Value
▶	Input Variable To Observe (Linked Value)	LINKED
	Imposed Maximum For Data Normalization - 'N/A' for not applicable (Optional -	N/A
	Imposed Minimum For Data Normalization - 'N/A' for not applicable (Optional -	N/A

Figure A6-5. Precipitation Condition Element Parameters

A6.2.3.1 Input Variable To Observe (Linked Value) Linked Code

```
Public Function F343(ob As CFCSim_ModelingElementInstance)

  F343 =  Elements("13")("Total")

End Function
```

A6.2.4 Element 4 Precipitation Seven Day Sum Condition Element


Condition ANN #346
Parameters
Outputs
Statistics
OK
Cancel
Print

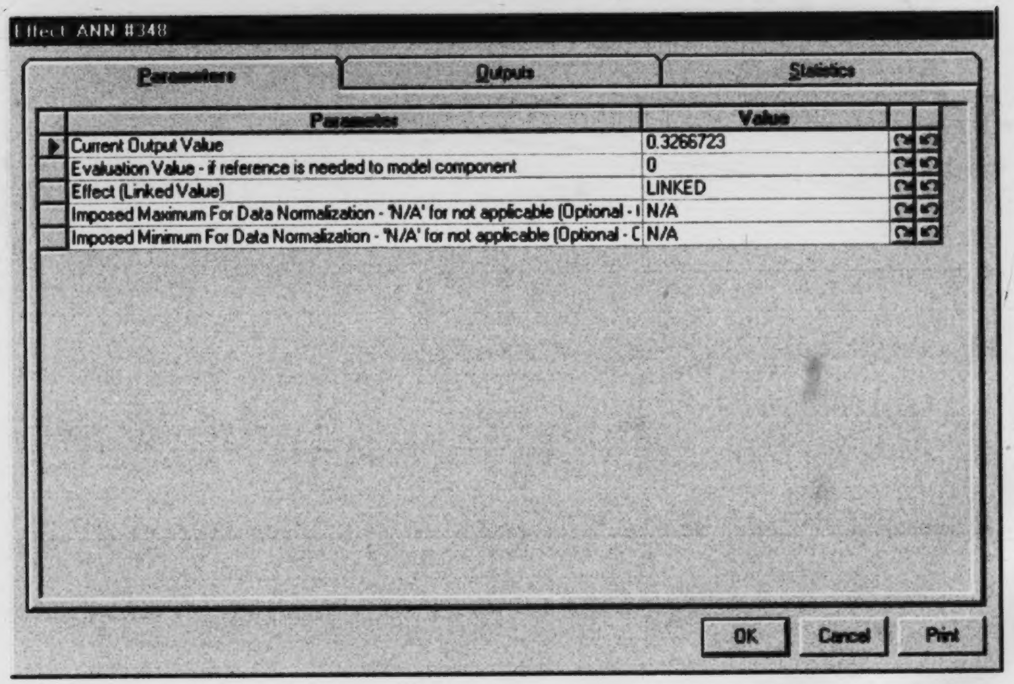
Parameter	Value
Input Variable To Observe (Linked Value)	LINKED
Imposed Maximum For Data Normalization - 'N/A' for not applicable (Optional - I	N/A
Imposed Minimum For Data Normalization - 'N/A' for not applicable (Optional - C	N/A

Figure A6-6. Precipitation Seven Day Sum Condition Element Parameters

A6.2.4.1 Input Variable To Observe (Linked Value) Linked Code

```
Public Function F347(ob As CFCSim_ModelingElementInstance)

  F347 =  Elements("14")("Total")

End Function
```

A6.2.5 Element 5 Production Factor Effect Element



Parameter	Value
Current Output Value	0.3266723
Evaluation Value - if reference is needed to model component	0
Effect (Linked Value)	LINKED
Imposed Maximum For Data Normalization - 'N/A' for not applicable (Optional - I	N/A
Imposed Minimum For Data Normalization - 'N/A' for not applicable (Optional - C	N/A

Figure A6-7. Production Factor Element Parameters

A6.2.5.1 Effect (Linked Value) Linked Code

```
Public Function F349(ob As CFCSim_ModelingElementInstance)

Elements("2")("ProductivityFactor").Distribution.ParameterValue(0) = ob("ObsValue")

F349 = 0

End Function
```

A6.3 Modified Task Element Programming Code

```
Option Explicit
```

```
Public Function MhActivity_OnCreate(ob As CFCSim_ModelingElementInstance, x As Single, y As Single) As Boolean
        ob.OnCreate x,y,True
        MhActivity_OnCreate=True

        ob.AddAttribute "TotalMhrs","Total Manhours",CFC_Distribution, CFC_Single, CFC_ReadWrite,0
        ob.AddAttribute "Productivity","Productivity (Manhours/TimeUnit)",CFC_Distribution, CFC_Single, CFC_ReadWrite,0
        ob.AddAttribute "ProductivityFactor","Productivity Factor",CFC_Distribution, CFC_Single, CFC_ReadWrite,0,1

        ob("TotalMhrs") = 40
        ob("Productivity") = 10
        ob("ProductivityFactor") = 1

        ob.AddAttribute "NumSrv","Number of Servers",CFC_Text,CFC_ListBox, CFC_ReadWrite,0
        ob("NumSrv")="Unconstrained"
        ob.AddAttribute
"Description","Description",CFC_Text,CFC_Single,CFC_ReadWrite
        ob("Description")="Task"& ob.Id

        ob.SetNumCoordinates 2
        ob.CoordinatesX(0)=x
        ob.CoordinatesY(0)=y
        ob.CoordinatesX(1)=x+100
        ob.CoordinatesY(1)=y+50

        ob.AddResource "Server",1

        ob.AddConnectionPoint "In" , x-10, y+25, CInput, 5
        ob.AddConnectionPoint "Out",x+115,y+25,COutput,5
End Function
```

```
Public Sub MhActivity_OnDragDraw(ob As CFCSim_ModelingElementInstance)
        ob.OnDraw
End Sub
```

```
Public Sub MhActivity_OnDraw(ob As CFCSim_ModelingElementInstance)
        CDC.ChangeFont "Courier New",13,True,False,False,False
        CDC.RenderPicture "Task.bmp",ob.CoordinatesX(0)+1,ob.CoordinatesY(0)+1,32,32
        CDC.Rectangle
ob.CoordinatesX(0),ob.CoordinatesY(0),ob.CoordinatesX(0)+33,ob.CoordinatesY(0)+33
        CDC.Rectangle
ob.CoordinatesX(0),ob.CoordinatesY(0),ob.CoordinatesX(1),ob.CoordinatesY(1)

        CDC.ChangeFont "Arial",11,True,False,False,False
        CDC.TextOut ob.CoordinatesX(0)+33,ob.CoordinatesY(0)+7, ob("Description")


        If ob("NumSrv")="Unconstrained" Then
                CDC.TextOut ob.CoordinatesX(0)+33,ob.CoordinatesY(0)+20,
"Unconstrained"
        Else
                CDC.TextOut ob.CoordinatesX(0)+40,ob.CoordinatesY(0)+20, "Constrained"
                CDC.TextOut ob.CoordinatesX(0)+45,ob.CoordinatesY(0)+35, " #Srv: "&
ob("NumSrv")
        End If

        If ob.Selected Then
                CDC.ChangeLineStyle CFC_DOT,1,RGB(255,0,0)
                CDC.Rectangle ob.CoordinatesX(0)-2,ob.CoordinatesY(0)-
2,ob.CoordinatesX(1)+2,ob.CoordinatesY(1)+2
```

```
        End If
        CDC.ChangeLineStyle CFC_SOLID,1,RGB(0,0,0)


        ob.DrawConnectionPoints
End Sub
```

```
Public Sub MhActivity_OnListBoxInitialize(ob As CFCSim_ModelingElementInstance, attr
As CFCSim_Attribute, lstList As Object)
        lstList.additem "Unconstrained"
End Sub
```

```
Public Sub MhActivity_OnSimulationInitialize(ob As CFCSim_ModelingElementInstance)
        ob.AddEvent "Welcome",True
        ob.AddEvent "Start"
        ob.AddEvent "StartWithRes"
        ob.AddEvent "Finish"
End Sub
```

```
Public Sub MhActivity_OnSimulationInitializeRun(ob As
CFCSim_ModelingElementInstance, RunNum As Integer)
        If ob("NumSrv")<>"Unconstrained" Then
                ob.res("Server").NumResources =ob("NumSrv")
        End If
End Sub
```

```
Public Sub MhActivity_OnSimulationProcessEvent(ob As
CFCSim_ModelingElementInstance, MyEvent As String, Entity As CFCSim_Entity)
    Dim Duration As Double
    Dim Guess As Double
    Dim SimDuration As Double

    Select Case MyEvent
        Case "Welcome"
            Entity("CummulativeMhrs") = 0
            ob.ScheduleEvent Entity,"Start",0
        Case "Start"
            If ob("NumSrv")="Unconstrained" Then
                Duration=ob("Productivity")*ob("ProductivityFactor")
                Guess = Duration + Entity("CummulativeMhrs")
                If Guess >= ob("TotalMhrs") Then
                    Duration = ob("TotalMhrs") -
Entity("CummulativeMhrs")
                    Entity("CummulativeMhrs") = 0
                    SimDuration = Duration /ob("ProductivityFactor")
                    ob.ScheduleEvent Entity,"Finish",SimDuration
                Else
```

```
                    Entity("CummulativeMhrs")=
Entity("CummulativeMhrs") + Duration
                    SimDuration = Duration /ob("ProductivityFactor")
                    ob.ScheduleEvent Entity,"Start",SimDuration
                End If
                Tracer.Trace "Entity: " & Entity.Id & " will accrue " &
Duration & " manhours in activity " & ob.Id & " over " & SimDuration & " time units"
,"Simulation"

            Else
                If ob.RequestResource ("Server",Entity) = True Then
                    Duration=ob("Productivity")*ob("ProductivityFactor")
                    Guess = Duration + Entity("CummulativeMhrs")
                    If Guess >= ob("TotalMhrs") Then
                        Duration = ob("TotalMhrs") -
Entity("CummulativeMhrs")
                        Entity("CummulativeMhrs") = 0
                        SimDuration = Duration
/ob("ProductivityFactor")
                        ob.ScheduleEvent Entity,"Finish",SimDuration
                    Else
                        Entity("CummulativeMhrs")=
Entity("CummulativeMhrs") + Duration
```

```
                        SimDuration = Duration
/ob("ProductivityFactor")
                        ob.ScheduleEvent
Entity,"StartWithRes",SimDuration
                    End If
                End If
                Tracer.Trace "Entity: " & Entity.Id & " will accrue " &
Duration & " manhours in activity " & ob.Id & " over " & SimDuration & " time units"
,"Simulation"
            End If

        Case "StartWithRes"
                Duration=ob("Productivity")*ob("ProductivityFactor")
                Guess = Duration + Entity("CummulativeMhrs")
                If Guess >= ob("TotalMhrs") Then
                    Duration = ob("TotalMhrs") -
Entity("CummulativeMhrs")
                    Entity("CummulativeMhrs") = 0
                    SimDuration = Duration /ob("ProductivityFactor")
                    ob.ScheduleEvent Entity,"Finish",SimDuration
                Else
                    Entity("CummulativeMhrs")=
Entity("CummulativeMhrs") + Duration
```

```
                    SimDuration = Duration /ob("ProductivityFactor")
                    ob.ScheduleEvent Entity,"StartWithRes",SimDuration
                End If
                Tracer.Trace "Entity: " & Entity.Id & " will accrue " &
Duration & " manhours in activity " & ob.Id & " over " & SimDuration & " time units"
,"Simulation"

        Case "Finish"
            If ob("NumSrv")<> "Unconstrained" Then
                ob.ReleaseResource ("Server",Entity)
            End If
            Tracer.Trace "Delay of Entity: " & Entity.Id & " Completed" ,
"Simulation"
            ob.TransferOut Entity
    End Select
End Sub
```

```
Public Function MhActivity_OnValidateParameters(ob As
CFCSim_ModelingElementInstance, Parameters As Object) As Boolean
    MhActivity_OnValidateParameters=ob.OnValidateParameters (Parameters,True)
    If MhActivity_OnValidateParameters=False Then Exit Function
    If Parameters("NumSrv") <>"Unconstrained" And Not
IsNumeric(Parameters("NumSrv")) Then
```

```
                MessagePrompt "Please enter an integer value for the number of servers or
select ""Unconstrained"""
                MhActivity_OnValidateParameters=False
        End If
End Function
```

APPENDIX 7 – Earthmoving Simulation Information

A7.1 Model Overview

The earthmoving simulation model is a single agent example application where the agent is employed to interpret weather data from an information source and apply the effects of weather through a productivity factor to the shovel production in the model. The model was constructed using the Common and Earthmoving Simulation templates (Hajjar and AbouRizk, 2002). The agent employed in this model makes use of the same weather neural network developed by Wales (1994) used in the bridge abutment construction simulation presented in Appendix 6. The following section provides information to construct the model.

A7.2 Model Construction

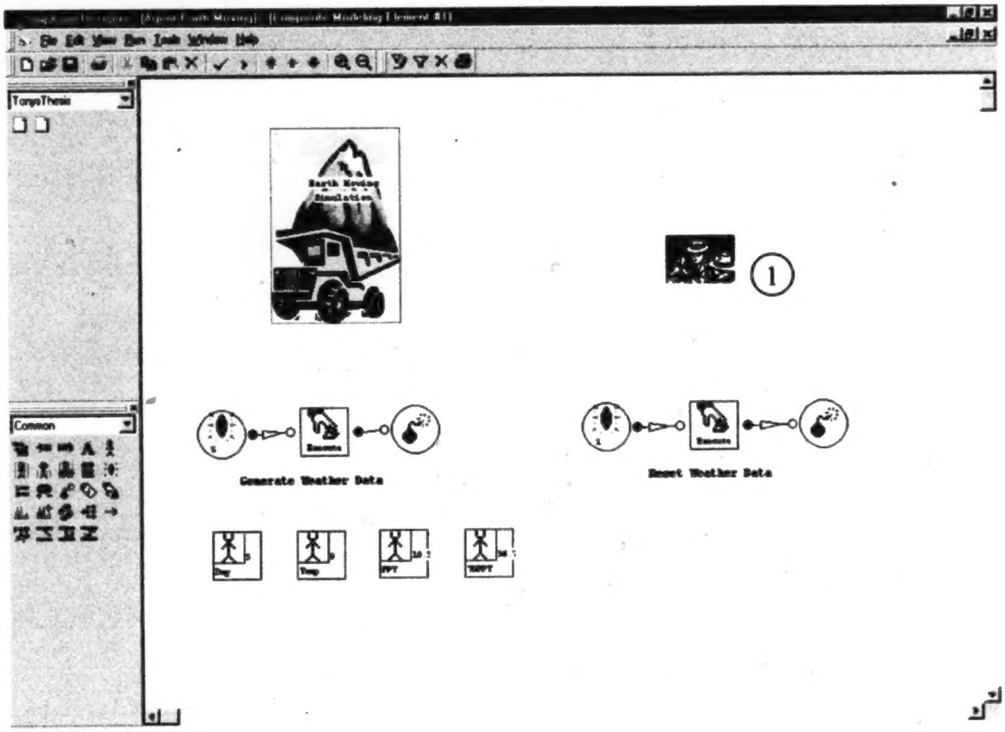

File Edit View Run Tools Window Help
TonyaThesis
Common
Earth Moving Simulation
①
Execute
Generate Weather Data
Execute
Reset Weather Data

Figure A7-1. Project Level View

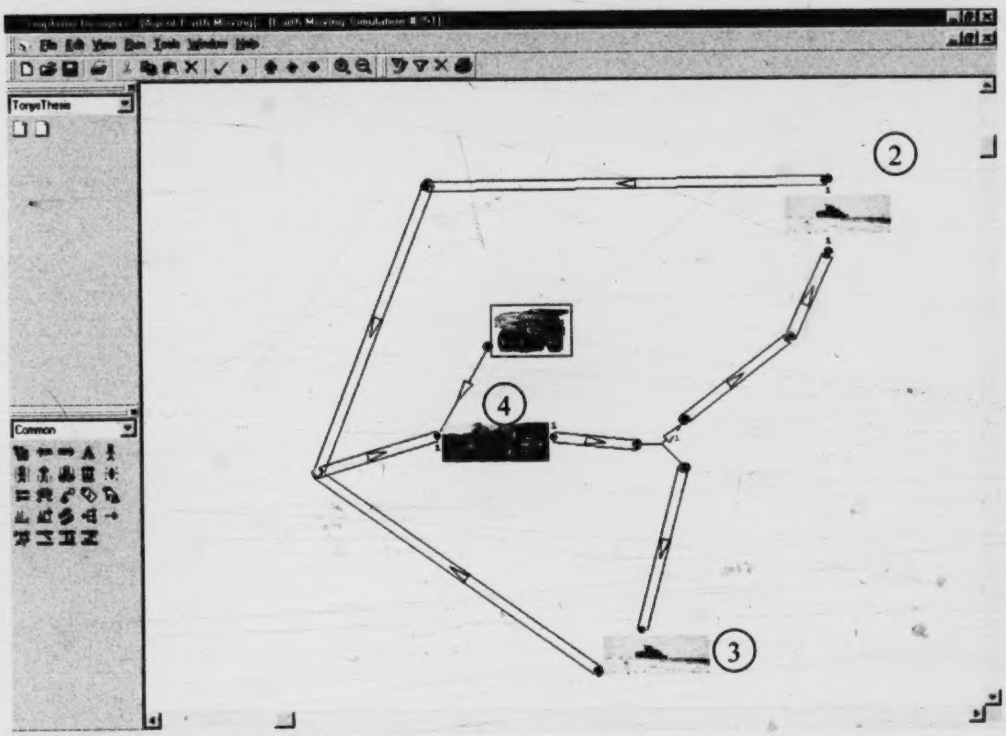

File
Edit
View
Run
Tools
Window
Help
TonyaThesis
Common
2
4
3

Figure A7-2. Activity Level View

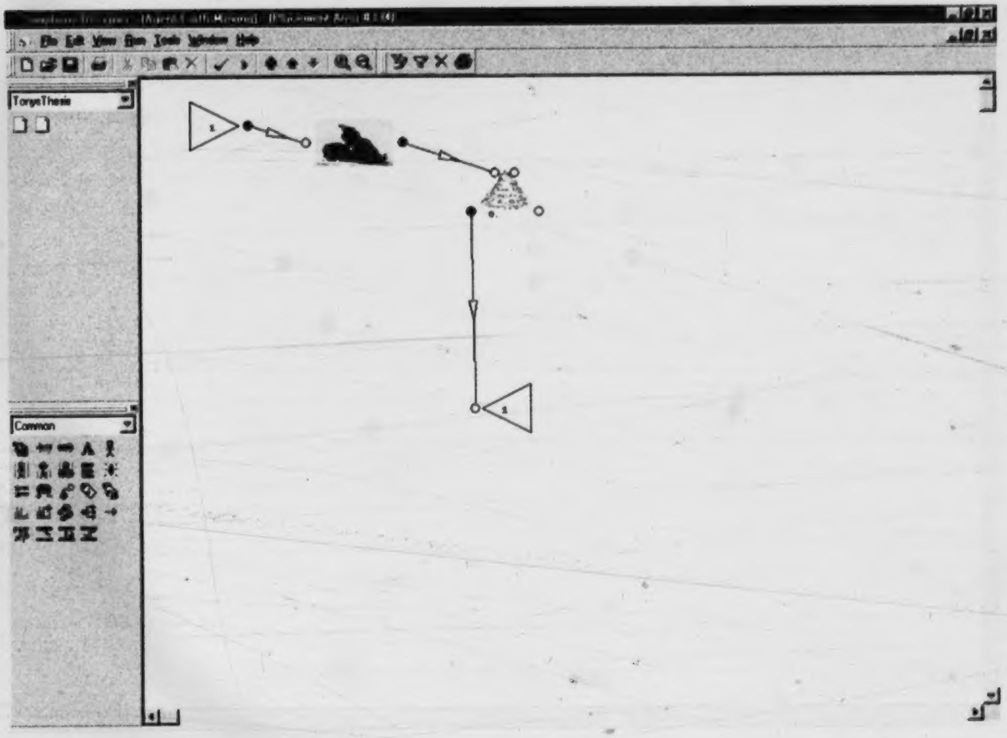

File
Edit
View
Run
Tools
Window
Help
TorgeThesis
Common
1
1

Figure A7-3. Dump Location 2 Child Element Window

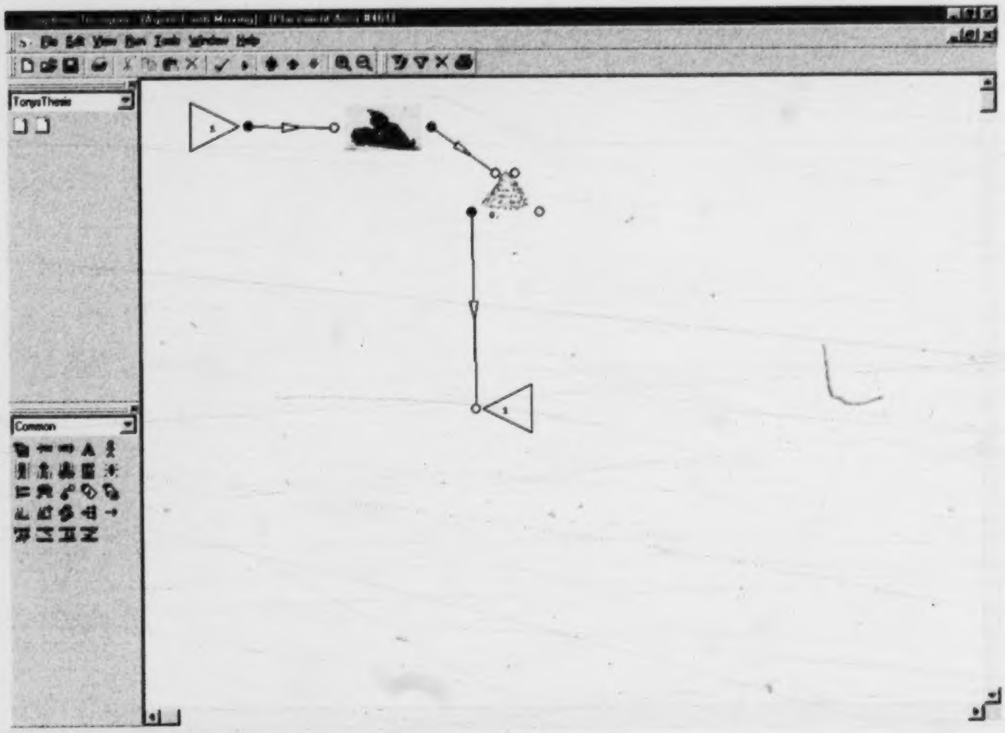

File Edit View Run Tools Window Help
TonysThesis
Common

Figure A7-4. Dump Location 3 Child Element Window

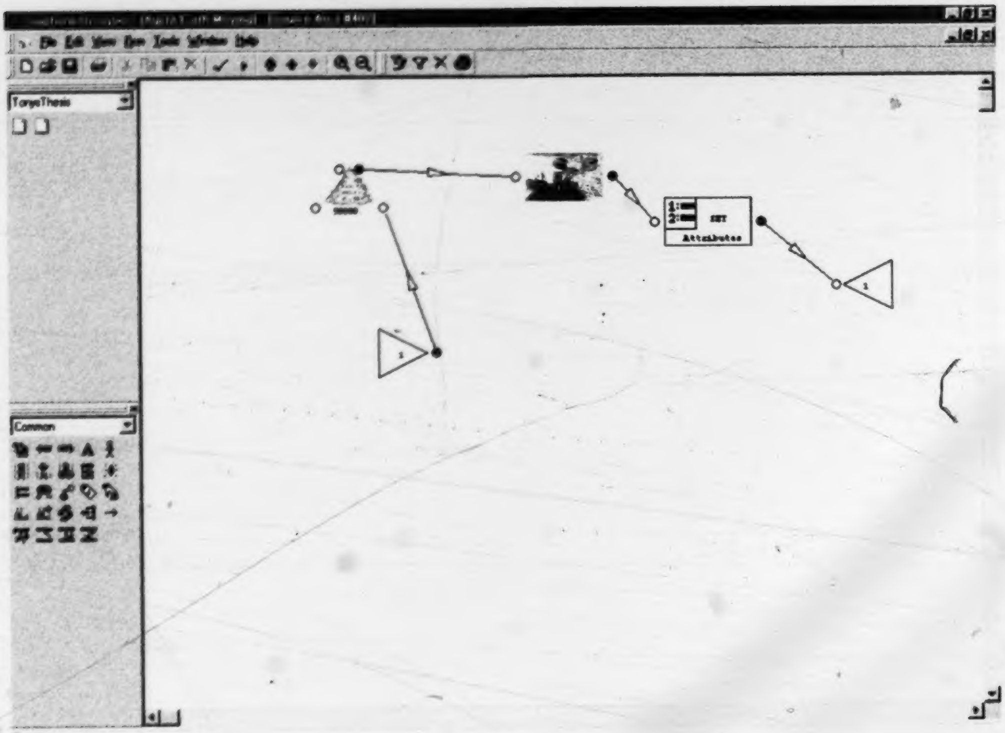
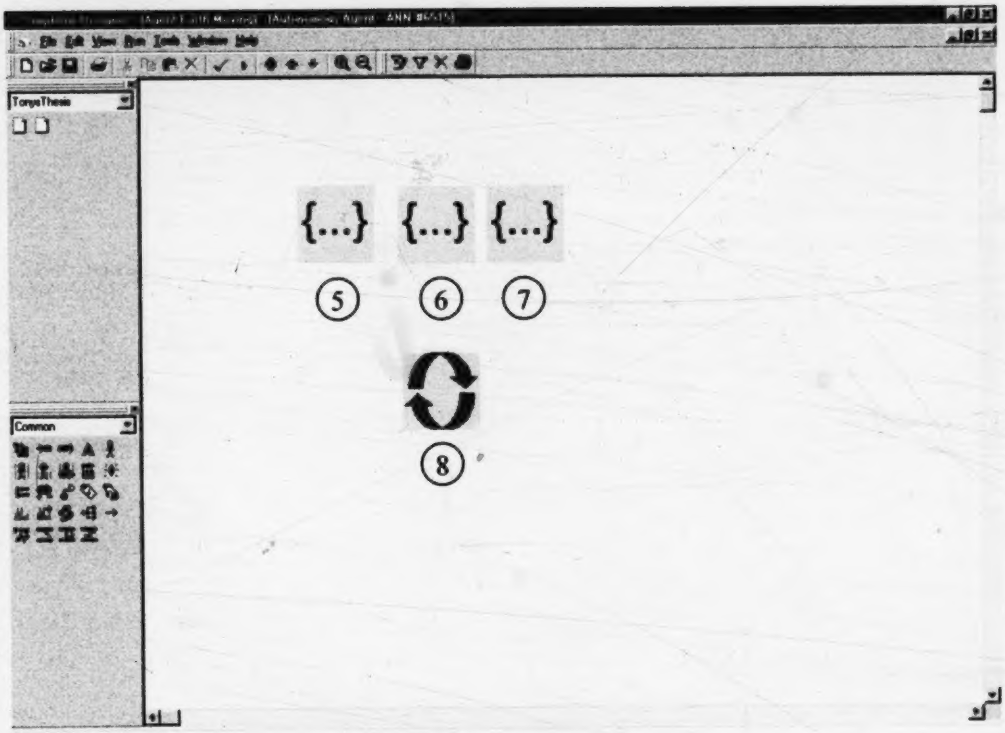

File Edit View Run Tools Window Help
TonyaThesis
Common
SET
Attributes
1
1

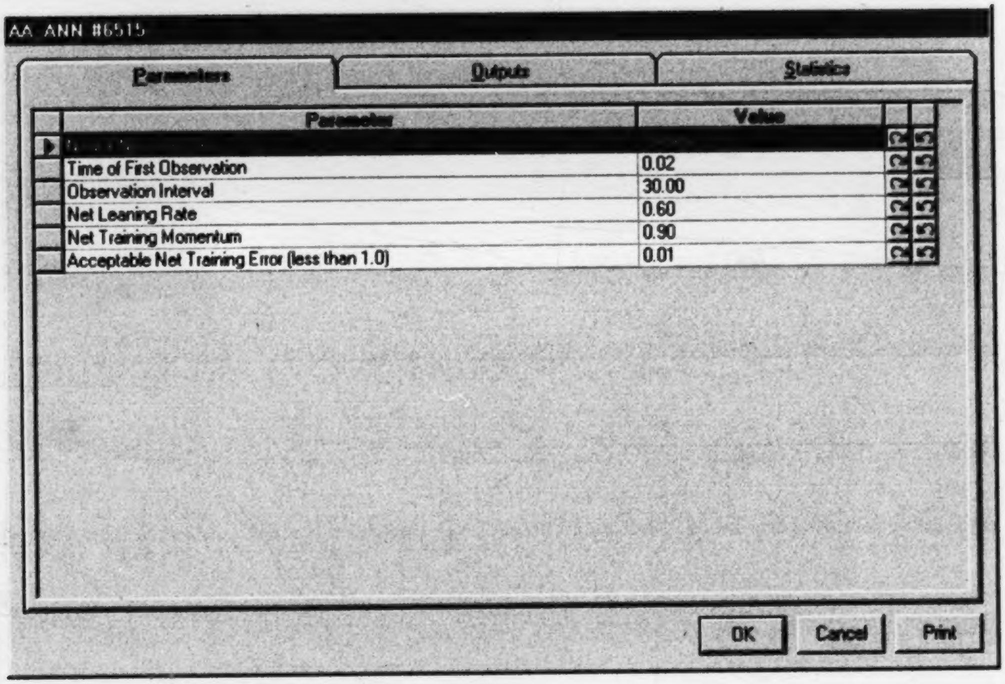
Figure A7-5. Source Location 4 Child Element Window



File
Edit
View
Run
Tools
Window
Help
TonyaThesis
Common
{...}
{...}
{...}
⑤
⑥
⑦
⑧

Figure A7-6. Agent Parent Element Child Window

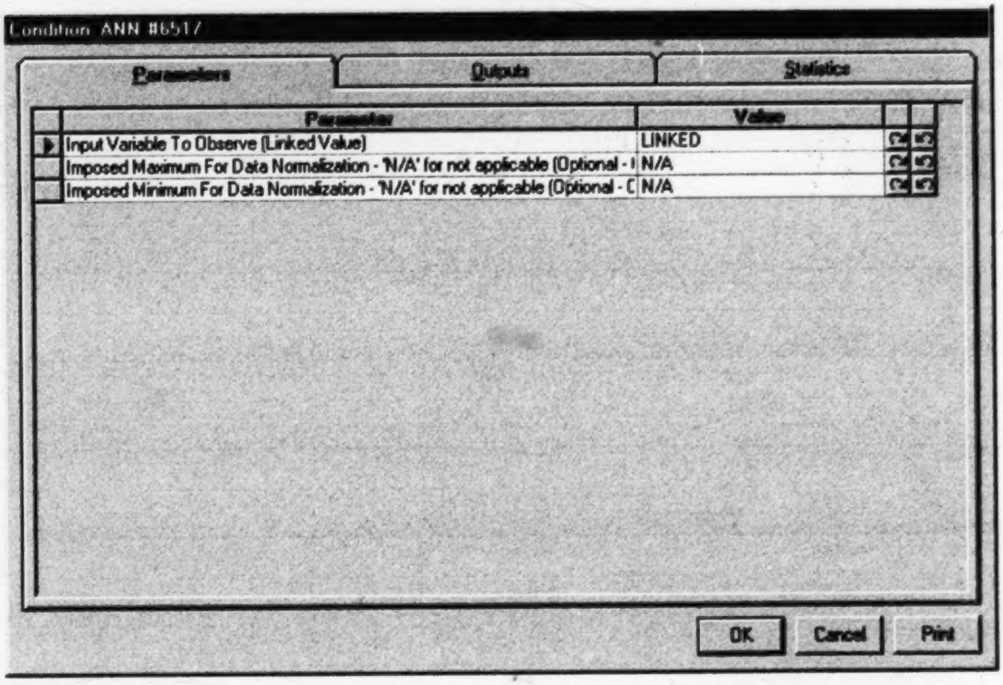
A7.2.1 Element 1 Agent Element


AA ANN #6515
Parameters
Outputs
Statistics
OK
Cancel
Print

Parameter	Value
[illegible]	
Time of First Observation	0.02
Observation Interval	30.00
Net Leaning Rate	0.60
Net Training Momentum	0.90
Acceptable Net Training Error (less than 1.0)	0.01

Figure A7-7. Agent Modeling Element Parameters

A7.2.2 Element 5 Average Daily Temperature Condition Element


Condition ANN #6517
Parameters
Outputs
Statistics
OK
Cancel
Print

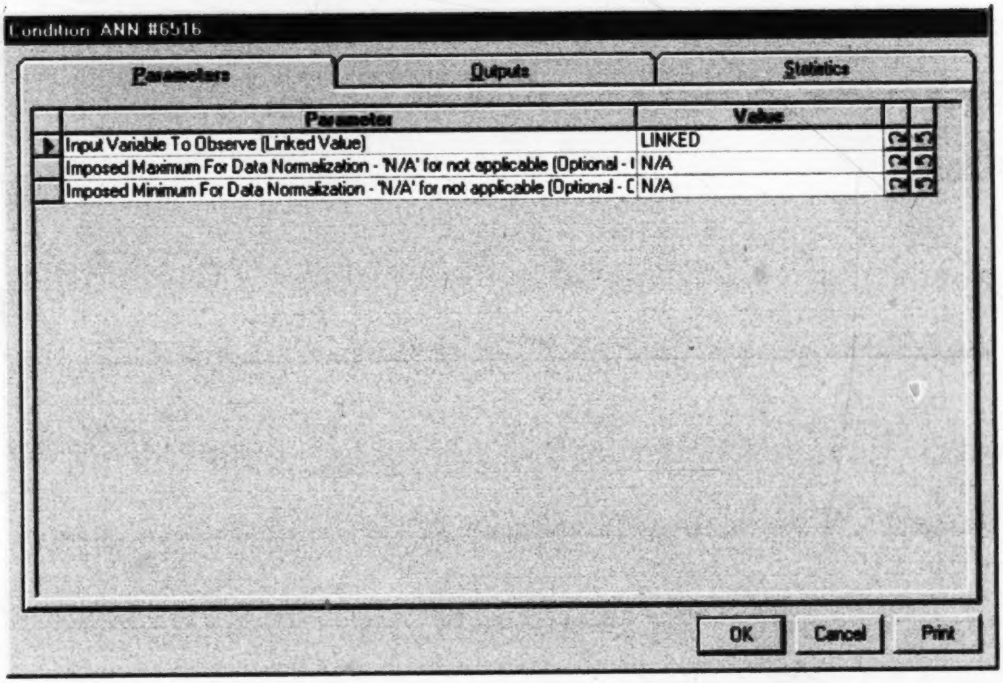
	Parameter	Value
▶	Input Variable To Observe (Linked Value)	LINKED
	Imposed Maximum For Data Normalization - 'N/A' for not applicable (Optional -	N/A
	Imposed Minimum For Data Normalization - 'N/A' for not applicable (Optional -	N/A

Figure A7-8. Average Daily Temperature Condition Modeling Element Parameters

A7.2.2.1 Input Variable To Observe (Linked Value) Linked Code

```
Public Function F6521(ob As CFCSim_ModelingElementInstance)
    F6521 = Elements("7")("Total")
End Function
```

A7.2.3 Element 6 Precipitation Condition Element


Condition ANN #6516
Parameters
Outputs
Statistics
OK
Cancel
Print

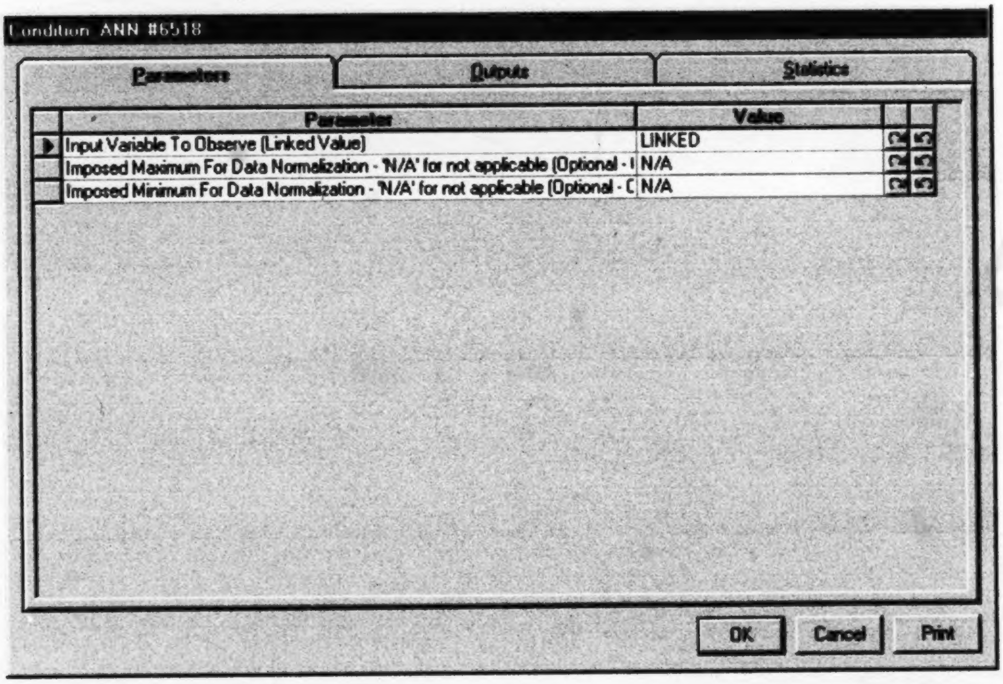
Parameter	Value
Input Variable To Observe (Linked Value)	LINKED
Imposed Maximum For Data Normalization - 'N/A' for not applicable (Optional -	N/A
Imposed Minimum For Data Normalization - 'N/A' for not applicable (Optional -	N/A

Figure A7-9. Precipitation Condition Modeling Element Parameters

A7.2.3.1 Input Variable To Observe (Linked Value) Linked Code

```
Public Function F6520(ob As CFCSim_ModelingElementInstance)

   F6520 =  Elements("8")("Total")

End Function
```

A7.2.4 Element 7 Precipitation Seven Day Sum Condition Element


Condition ANN #6518
Parameters
Outputs
Statistics
OK
Cancel
Print

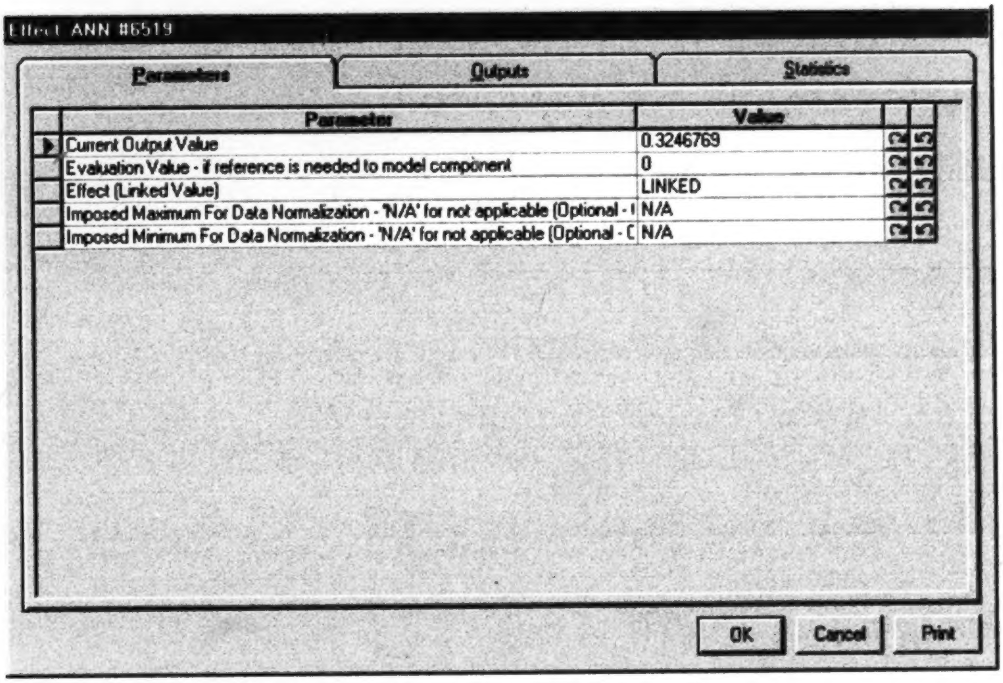
	Parameter	Value
▶	Input Variable To Observe (Linked Value)	LINKED
	Imposed Maximum For Data Normalization - 'N/A' for not applicable (Optional - I	N/A
	Imposed Minimum For Data Normalization - 'N/A' for not applicable (Optional - C	N/A

Figure A7-10. Precipitation Seven Day Sum Condition Modeling Element Parameters

A7.2.4.1 Input Variable To Observe (Linked Value) Linked Code

```
Public Function F6522(ob As CFCSim_ModelingElementInstance)

    F6522 =  Elements("9")("Total")

End Function
```

A7.2.5 Element 8 Production Factor Effect Element



Parameter	Value
Current Output Value	0.3246769
Evaluation Value - if reference is needed to model component	0
Effect (Linked Value)	LINKED
Imposed Maximum For Data Normalization - 'N/A' for not applicable (Optional -	N/A
Imposed Minimum For Data Normalization - 'N/A' for not applicable (Optional -	N/A

Figure A7-11. Production Factor Effect Modeling Element Parameters

A7.2.5.1 Effect (Linked Value) Linked Code

```
Public Function F6523(ob As CFCSim_ModelingElementInstance)
  Elements("35")("Productivity").Distribution.ParameterValue(0) = 2400*ob("ObsValue")
  F6523 = 0
End Function
```

APPENDIX 8 – North Edmonton Sanitary Trunk Tunnel Simulation Information

A8.1 Model Overview

The North Edmonton Sanitary Trunk (NEST) tunnel construction simulation model demonstrates the agent modeling framework developed through this research. The simulation model was constructed using the Common (Hajjar and AbouRizk, 2002) and Tunnel Construction (Van Tol et al., 2002). The following section provides the information required to construct the NEST tunnel construction simulation and embedding the agent framework within the modeling environment.

A8.2 Model Construction

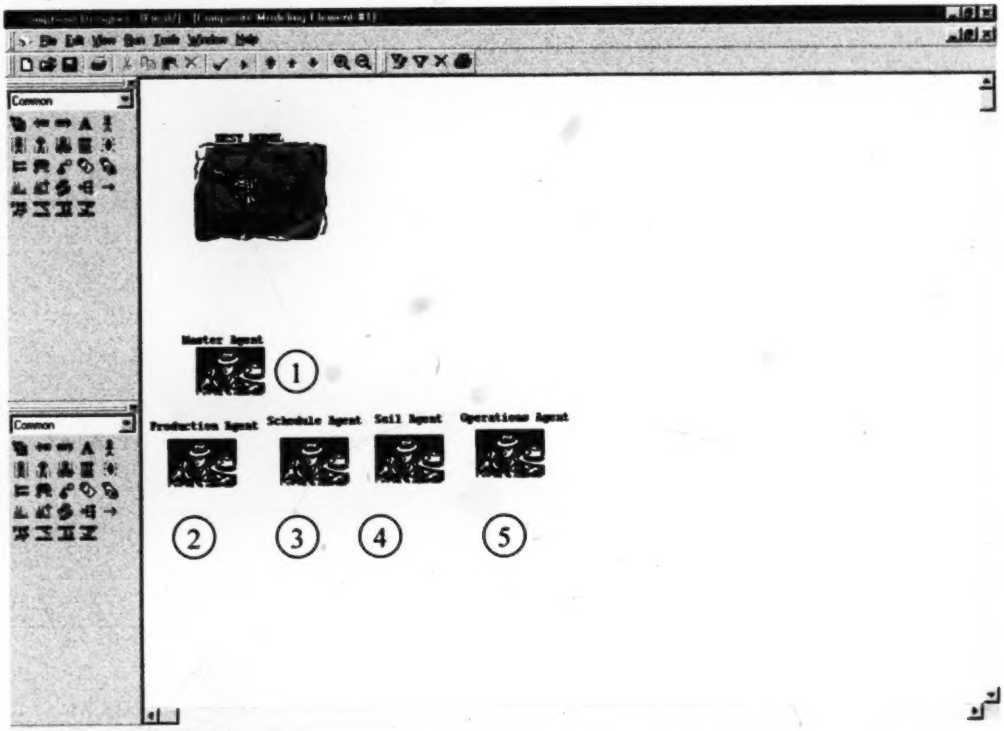

File Edit View Run Tools Window Help
Common
TEST MODEL
Master Agent
①
Production Agent
Schedule Agent
Soil Agent
Operations Agent
②
③
④
⑤
Common

Figure A8-1. Project Level View

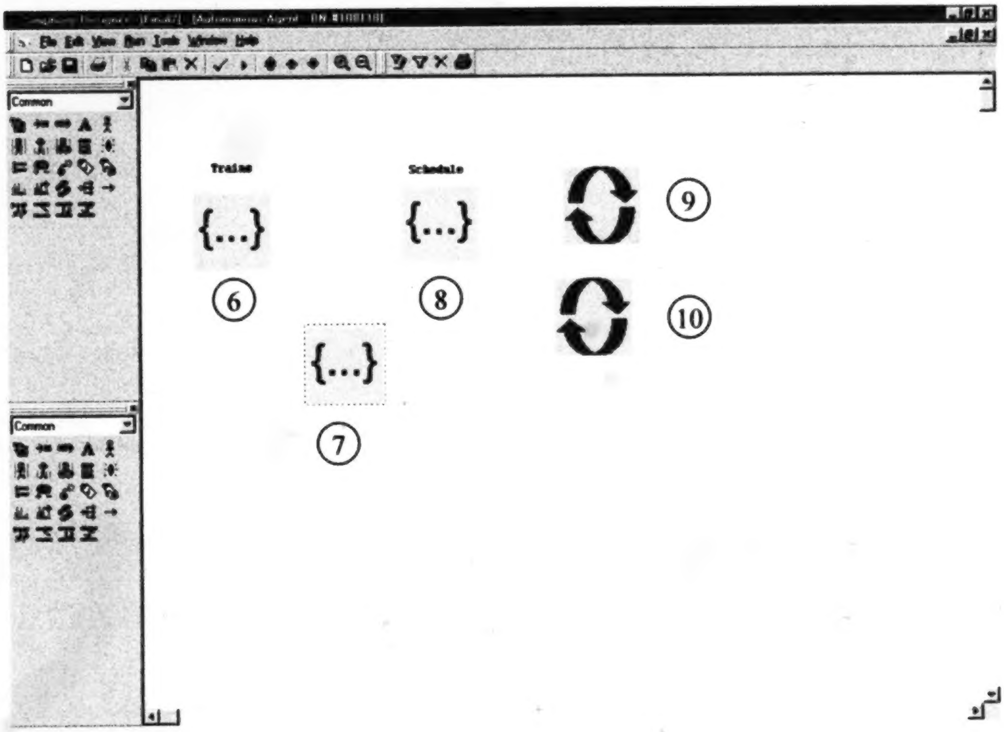

File Edit View Run Tools Window Help
Common
Common
Trains
Schedule
6
7
8
9
10

Figure A8-2. Master Agent Element 1 Child Window

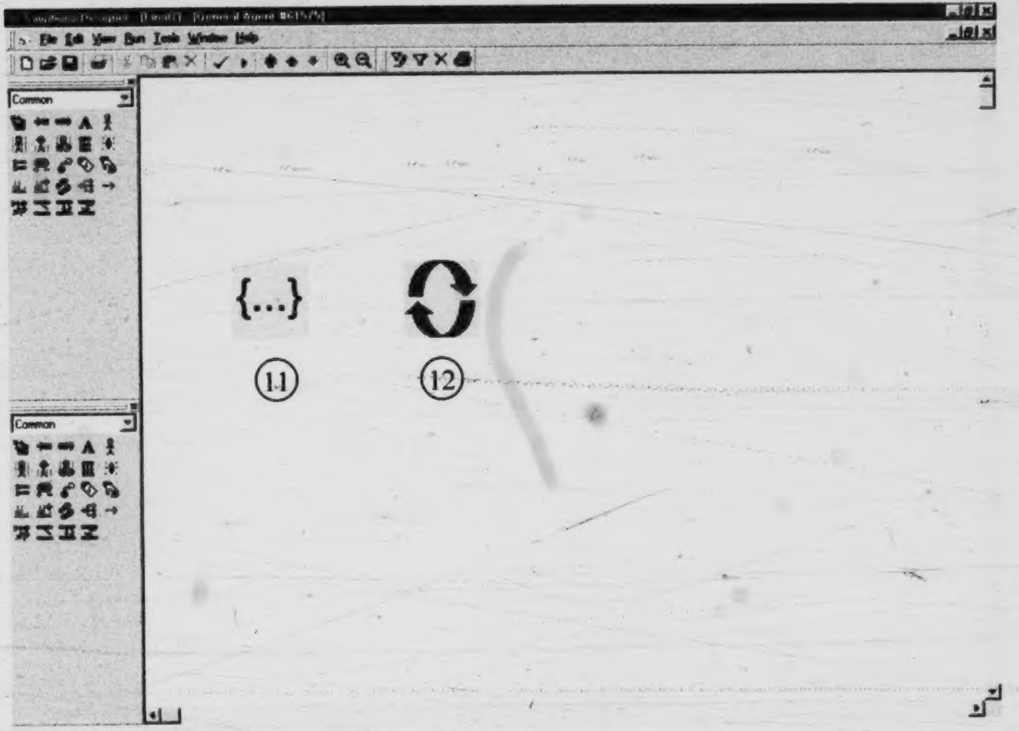

File Edit View Run Tools Window Help
Common
Common
{...}
⑪
⑫

Figure A8-3. Slave Agent Element 2 Child Window

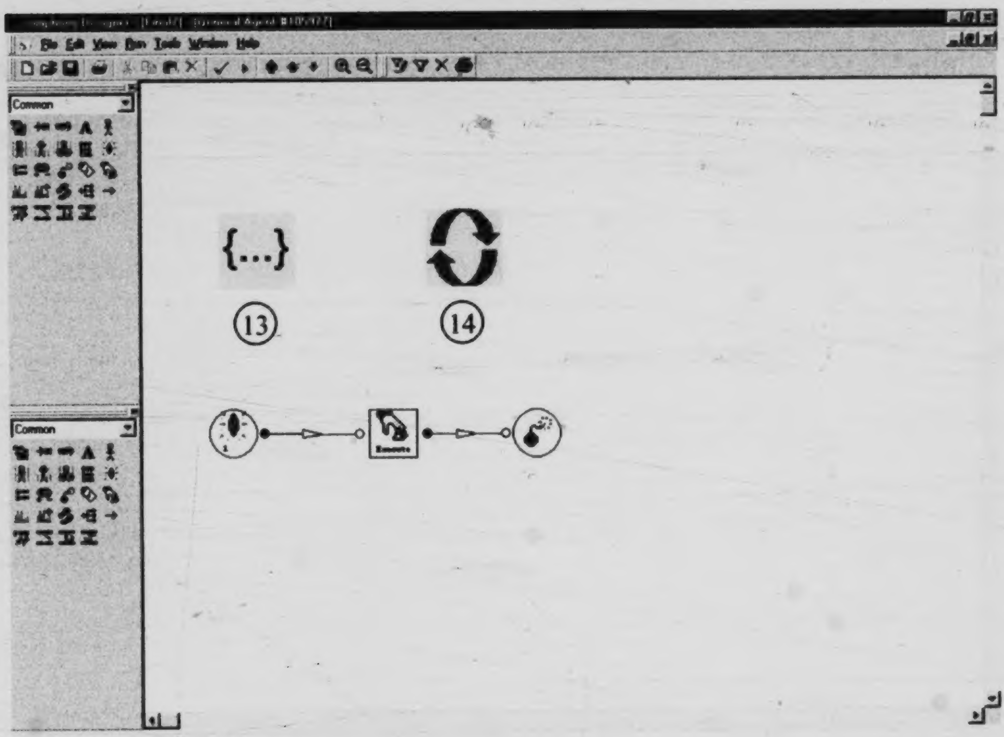

File Edit View Run Tools Window Help
Common
Common
{...}
13
14

Figure A8-4. Slave Agent Element 3 Child Window

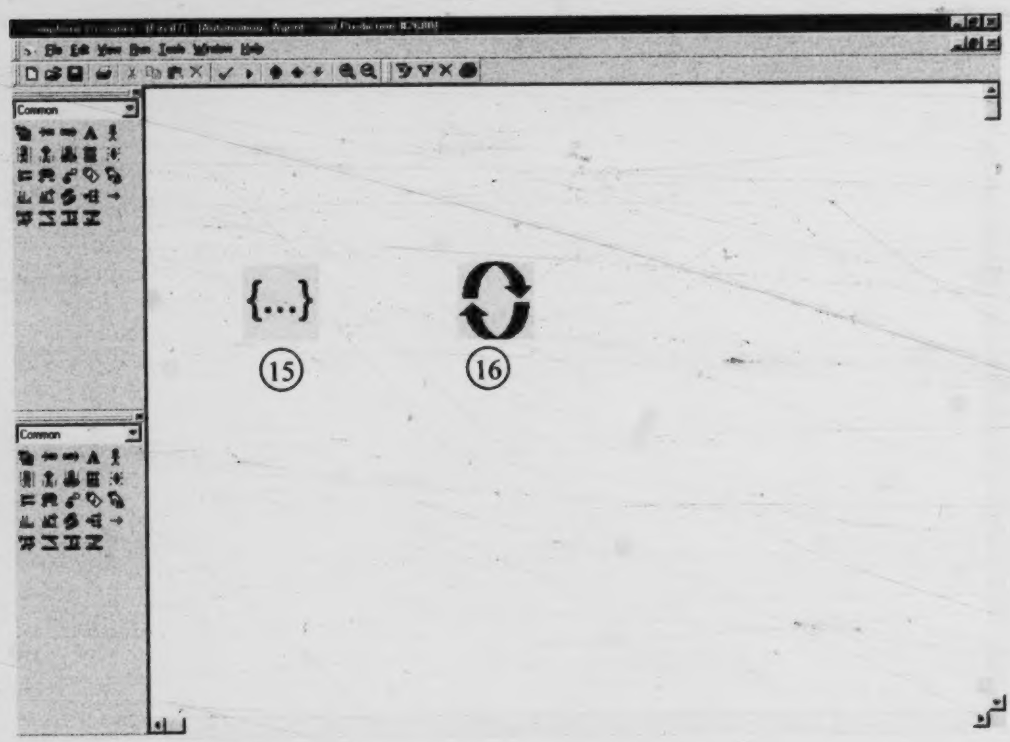

File Edit View Run Tools Window Help
Common
Common
{...}
⑮
⑯

Figure A8-5. Slave Agent Element 4 Child Window

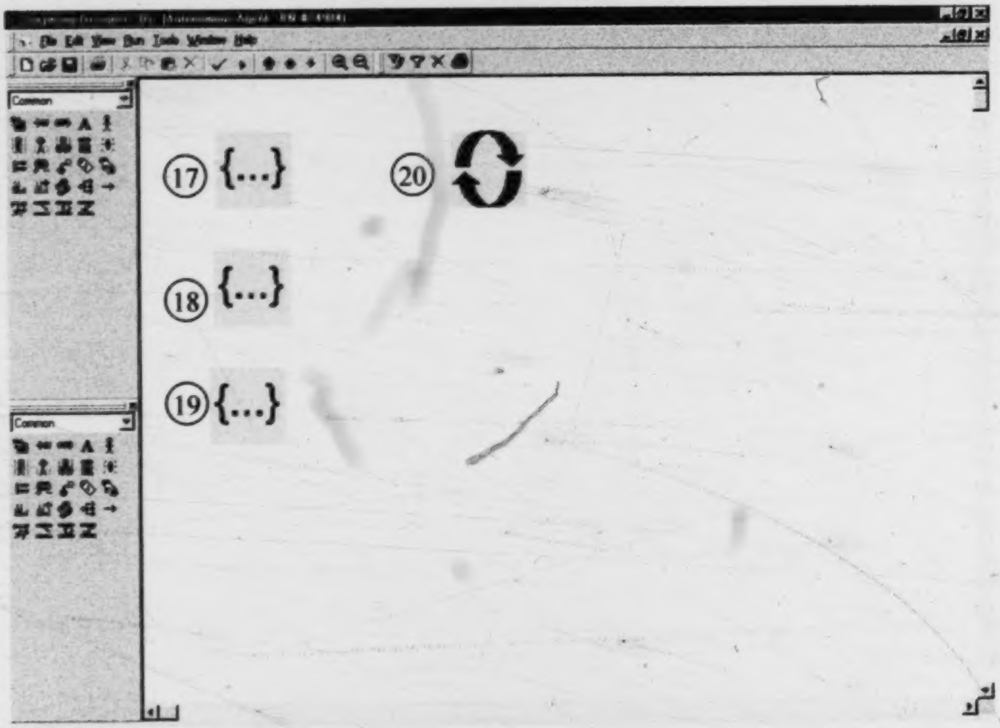
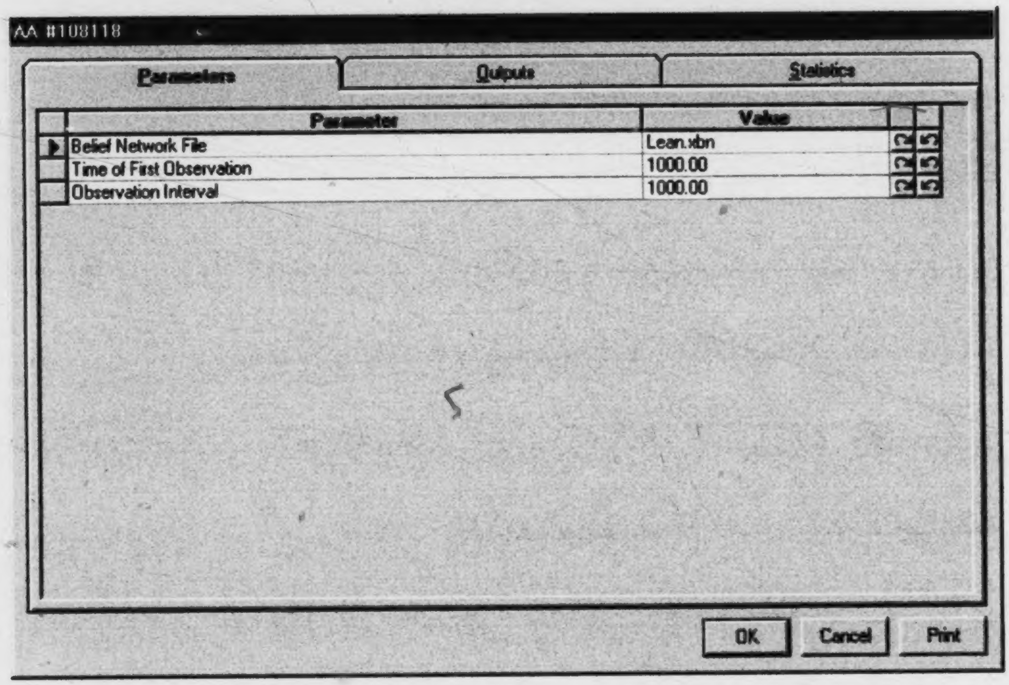

File Edit View Run Tools Window Help
Common
Common
⑰ {...}
⑱ {...}
⑲ {...}
⑳

Figure A8-6. Slave Agent Element 5 Child Window

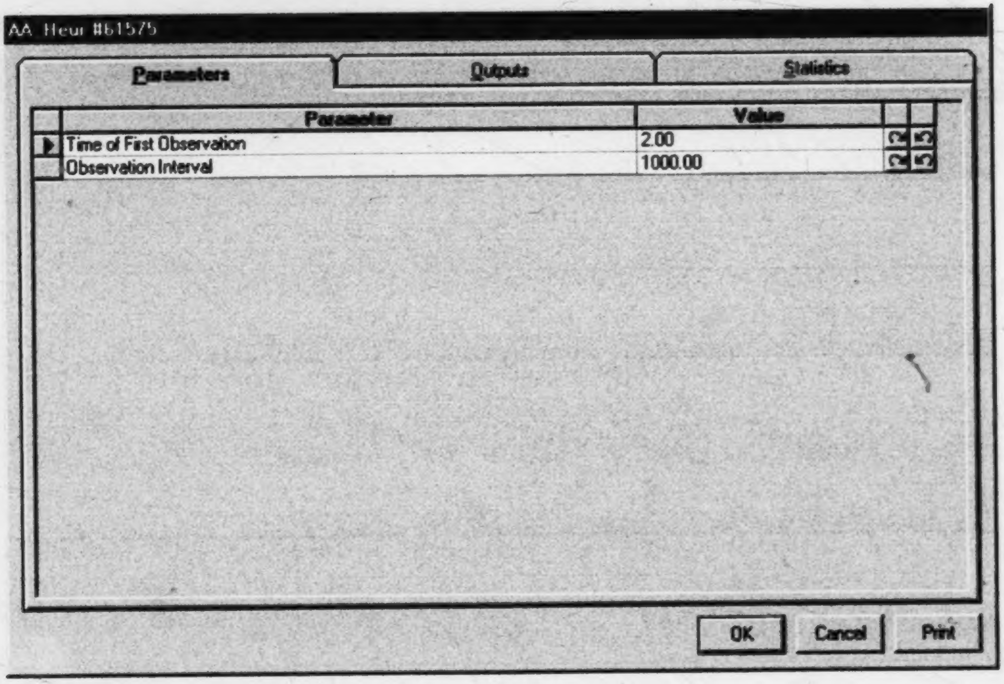
A8.2.1 Element 1 Belief Network Agent Element


AA #108118
Parameters
Outputs
Statistics
OK
Cancel
Print

Parameter	Value
Belief Network File	Lean.xbn
Time of First Observation	1000.00
Observation Interval	1000.00

Figure A8-7. Belief Network Agent Modeling Element Parameters

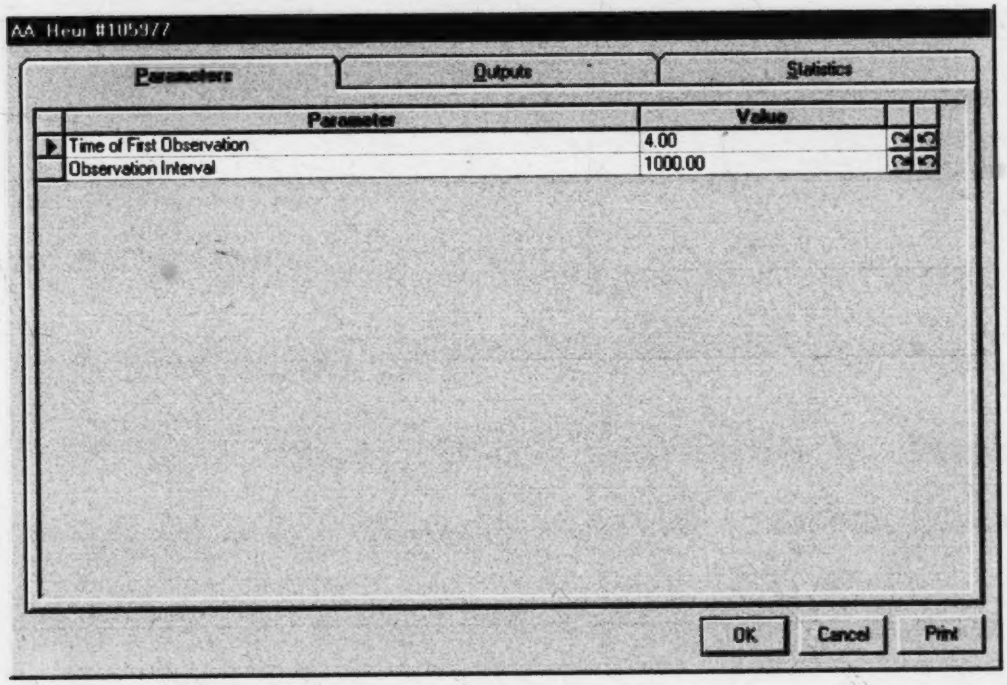
A8.2.2 Element 2 Rule Based Agent Element


AA Heur #61575
Parameters
Outputs
Statistics
OK
Cancel
Print

Parameter	Value
Time of First Observation	2.00
Observation Interval	1000.00

Figure A8-8. Rule Based Agent Modeling Element Parameters

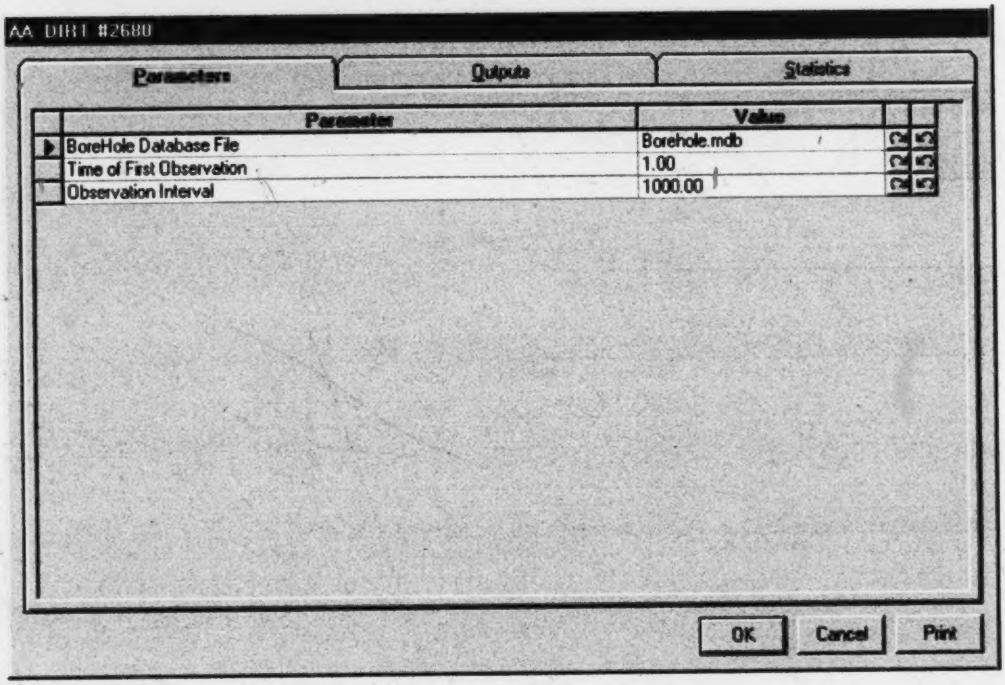
A8.2.3 Element 3 Rule Based Agent Element


AA Heur #105977
Parameters
Outputs
Statistics
OK
Cancel
Print

Parameter	Value
Time of First Observation	4.00
Observation Interval	1000.00

Figure A8-9. Rule Based Agent Modeling Element Parameters

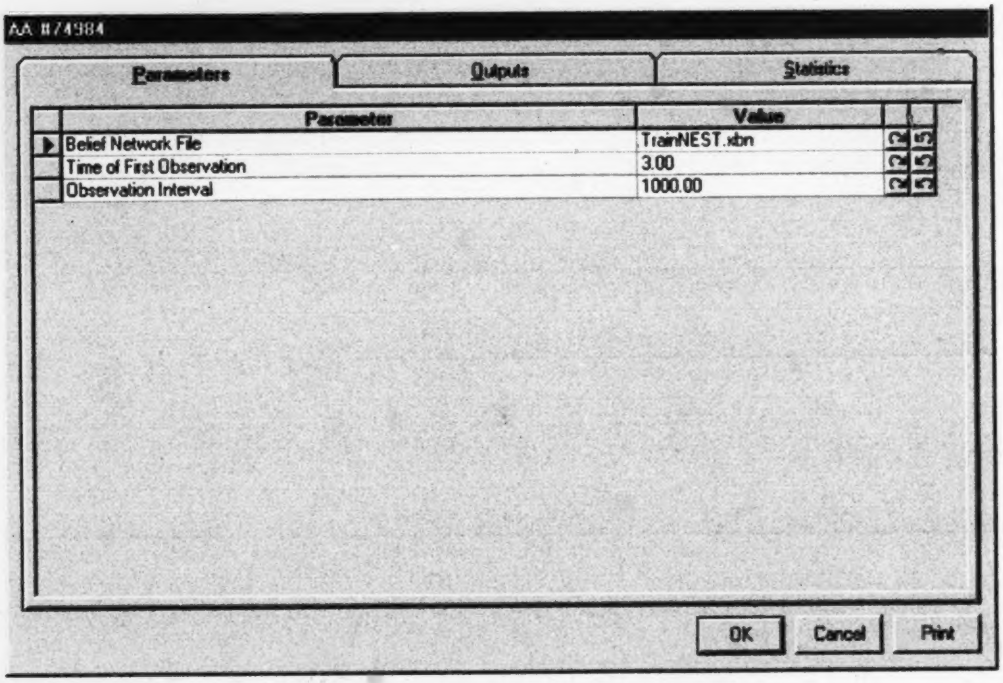
A8.2.4 Element 4 Algorithm Agent Element


AA DIRT #2680
Parameters
Outputs
Statistics
OK
Cancel
Print

Parameter	Value
BoreHole Database File	Borehole.mdb
Time of First Observation	1.00
Observation Interval	1000.00

Figure A8-10. Algorithm Agent Modeling Element Parameters

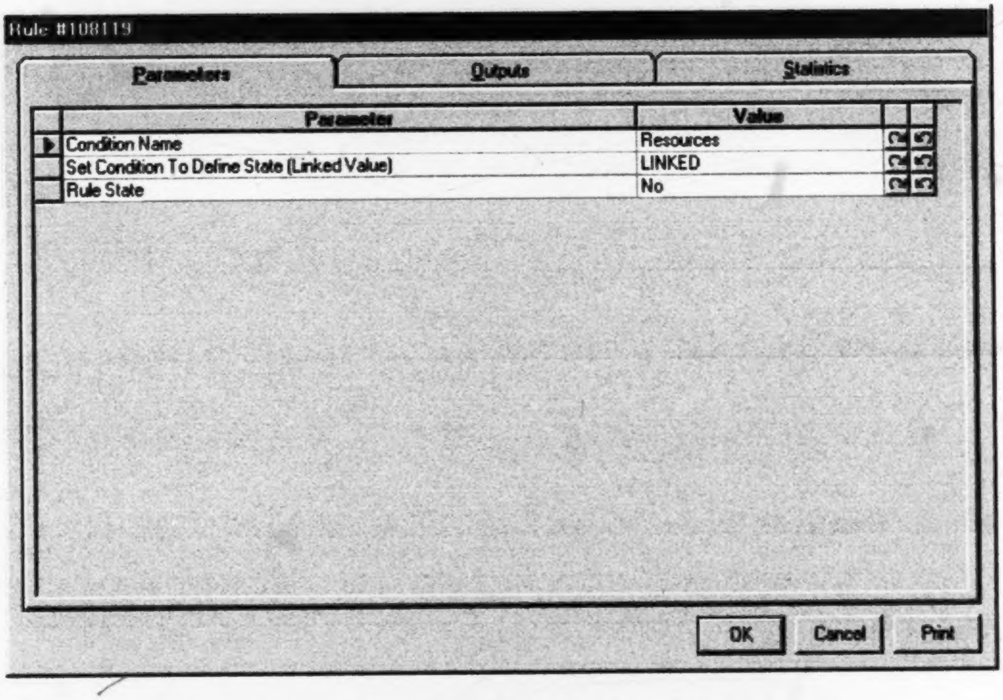
A8.2.5 Element 5 Belief Network Agent Element


AA #74984
Parameters
Outputs
Statistics
OK
Cancel
Print

Parameter	Value
Belief Network File	TrainNEST.xbn
Time of First Observation	3.00
Observation Interval	1000.00

Figure A8-11. Belief Network Agent Modeling Element Parameters

A8.2.6 Element 6 Belief Network Condition Element



Parameter	Value
Condition Name	Resources
Set Condition To Define State (Linked Value)	LINKED
Rule State	No

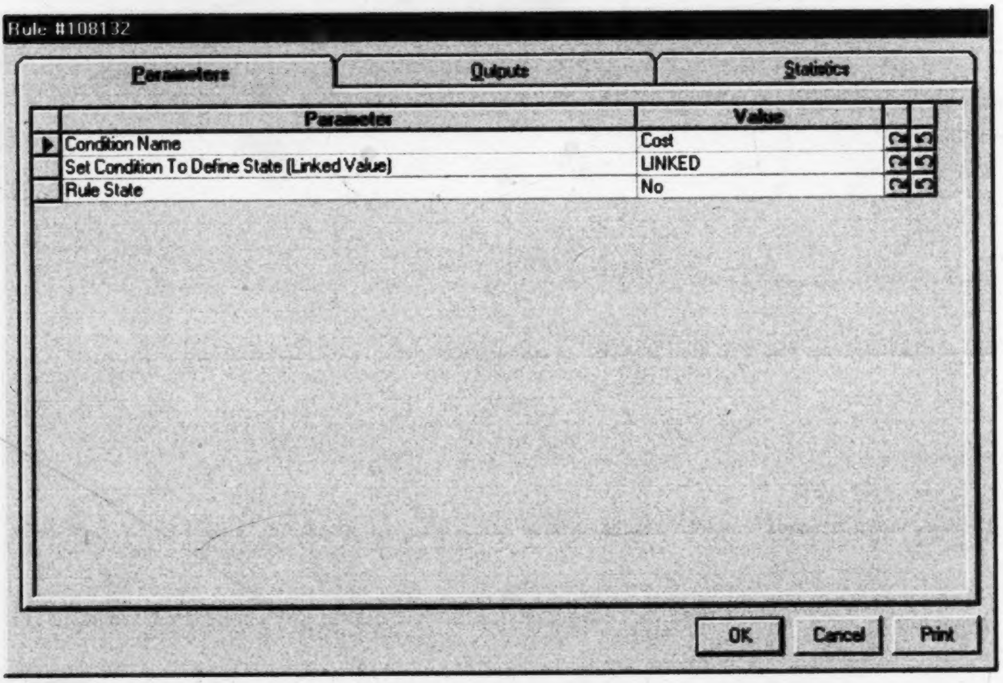
Figure A8-12. Belief Network Condition Modeling Element Parameters

A8.2.6.1 Set Condition To Define State (Linked Value) Linked Code

```
Public Function F108124(ob As CFCSim_ModelingElementInstance)

    Dim FindEle As CFCSim_ModelingElementInstance
    Dim NextEle As CFCSim_ModelingElementInstance
    Dim ThirdEle As CFCSim_ModelingElementInstance
    Dim Counter As Integer
    Dim TrainCount As String
    TrainCount = "Empty"
    For Each FindEle In ob.Parent.Parent.ChildElements
        If FindEle.ElementType = "CEM_Tunnel_Root" Then
            For Each NextEle In FindEle.ChildElements
                Counter = 0
                If NextEle.ElementType = "CEM_Tunnel_Undercut" Then
                    For Each ThirdEle In NextEle.ChildElements
                        If
ThirdEle.ElementType="CEM_Tunnel_Trains" Then Counter = Counter + 1
                    Next ThirdEle
                End If
                If Counter > 1 Then TrainCount = "Full"
            Next NextEle
        End If
    Next FindEle

    If TrainCount <> "Full" Then
        If   Elements("179")("CurrentBelief")   >=   Elements("179")("BeliefValue")
Then
            ob("RuleState") = "Yes"
        ElseIf  Elements("179")("CurrentBelief")  <  Elements("179")("BeliefValue")
Then
```

```
            ob("RuleState") = "No"
        Else
            ob("RuleState") = "Unobserved"
        End If
    End If

  F108124 = 0
End Function
```

A8.2.7 Element 7 Belief Network Condition Element


Rule #108132
Parameters
Outputs
Statistics
OK
Cancel
Print

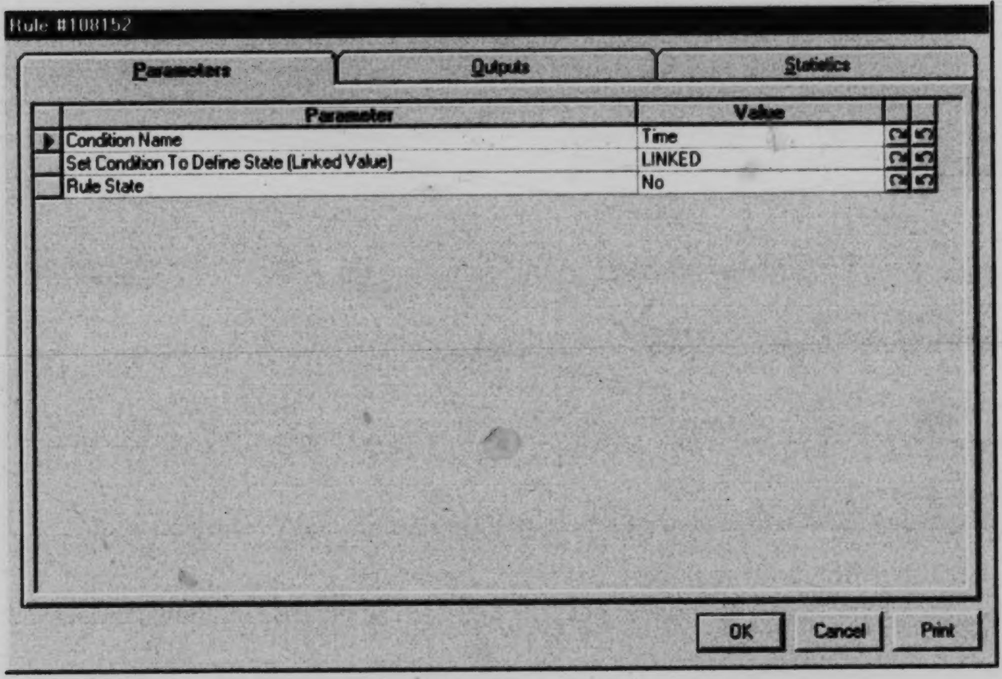
Parameter	Value
Condition Name	Cost
Set Condition To Define State (Linked Value)	LINKED
Rule State	No

Figure A8-13. Belief Network Condition Modeling Element Parameters

A8.2.7.1 Set Condition To Define State (Linked Value) Linked Code

```
Public Function F108133(ob As CFCSim_ModelingElementInstance)


    Dim FindEle As CFCSim_ModelingElementInstance

    Dim NextEle As CFCSim_ModelingElementInstance

    Dim ThirdEle As CFCSim_ModelingElementInstance

    Dim Counter As Integer

    Dim TrainCount As String


    TrainCount = "Empty"

    For Each FindEle In ob.Parent.Parent.ChildElements

        If FindEle.ElementType = "CEM_Tunnel_Root" Then

            For Each NextEle In FindEle.ChildElements

                Counter = 0

                If NextEle.ElementType = "CEM_Tunnel_Undercut" Then

                    For Each ThirdEle In NextEle.ChildElements

                        If

ThirdEle.ElementType="CEM_Tunnel_Trains" Then Counter = Counter + 1

                    Next ThirdEle

                End If

                If Counter > 1 Then TrainCount = "Full"

            Next NextEle

        End If
    Next FindEle

    If TrainCount <> "Full" And Elements("179")("CurrentBelief") >=
Elements("179")("BeliefValue") Then
        ob("RuleState") = "Yes"
    ElseIf Elements("170")("Evaluation") = "SecondShiftNeeded" Then
        ob("RuleState") = "Yes"
    ElseIf TrainCount <> "Full" And Elements("179")("CurrentBelief") <
Elements("179")("BeliefValue") Then
        ob("RuleState") = "No"
    ElseIf Elements("170")("Evaluation") <> "SecondShiftNeeded" Then
        ob("RuleState") = "No"
    Else
        ob("RuleState") = "Unobserved"
    End If
    F108133 = 0

End Function
```

A8.2.8 Element 8 Belief Network Condition Element


Rule #108152
Parameters
Outputs
Statistics
OK
Cancel
Print

Parameter	Value
Condition Name	Time
Set Condition To Define State (Linked Value)	LINKED
Rule State	No

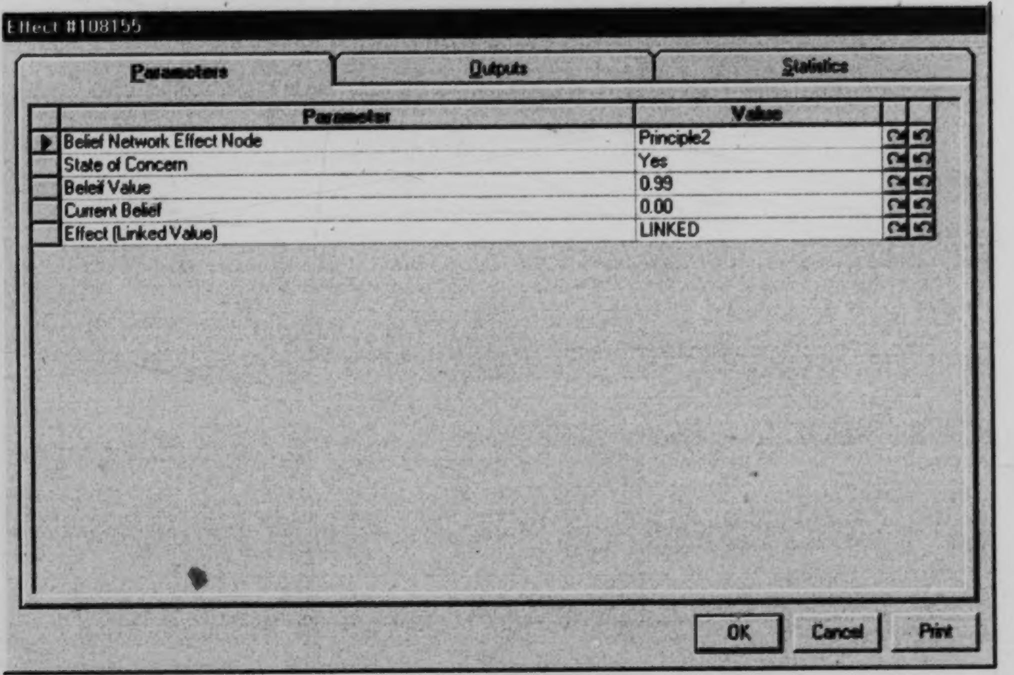
Figure A8-14. Belief Network Condition Modeling Element Parameters

A8.2.8.1 Set Condition To Define State (Linked Value) Linked Code

```
Public Function F108153(ob As CFCSim_ModelingElementInstance)


    If Elements("170")("Evaluation") = "SecondShiftNeeded" Then

        ob("RuleState") = "Yes"

    ElseIf Elements("170")("Evaluation") <> "SecondShiftNeeded" Then

        ob("RuleState") = "No"

    Else
```

```
            ob("RuleState") = "Unobserved"

        End If


        F108153 = 0


End Function
```

A8.2.9 Element 9 Belief Network Effect Element


Effect #108155
Parameters
Outputs
Statistics
OK
Cancel
Print

Parameter	Value
Belief Network Effect Node	Principle2
State of Concern	Yes
Beleif Value	0.99
Current Belief	0.00
Effect (Linked Value)	LINKED

Figure A8-15. Belief Network Effect Modeling Element Parameters

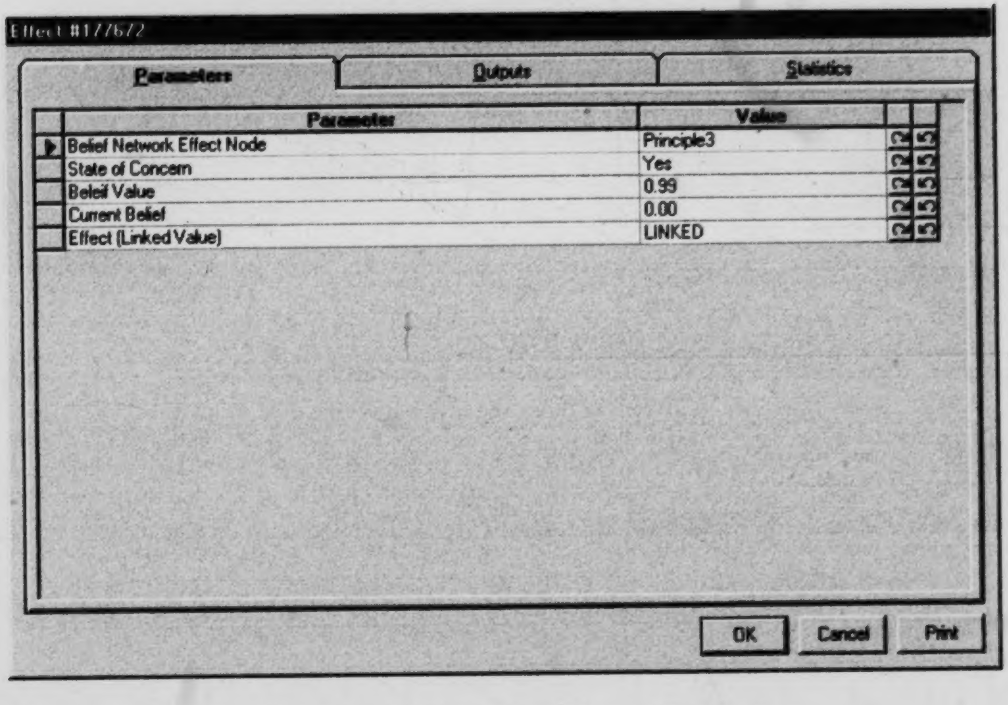
A8.2.9.1 Effect (Linked Value) Linked Code

```
Public Function F108156(ob As CFCSim_ModelingElementInstance)

    Dim FindEle As CFCSim_ModelingElementInstance
    Dim NextEle As CFCSim_ModelingElementInstance
    Dim ThirdEle As CFCSim_ModelingElementInstance
    Dim Counter As Integer
    Dim NEWTrain As CFCSim_ModelingElementInstance
    Dim Rel As CFCSim_Relation
    Dim FourthEle As CFCSim_ModelingElementInstance
    Dim FifthEle As CFCSim_ModelingElementInstance

    If SimTime > 5000 Then
        For Each FindEle In ob.Parent.Parent.ChildElements
            If FindEle.ElementType = "CEM_Tunnel_Root" Then
                For Each NextEle In FindEle.ChildElements
                    Counter = 0
                    If  NextEle.ElementType  =  "CEM_Tunnel_Undercut"
Then
                        For Each ThirdEle In NextEle.ChildElements
                            If
ThirdEle.ElementType="CEM_Tunnel_Trains" Then Counter = Counter + 1
```

```
                Next ThirdEle
            End If
            If Counter >0 Then GoTo NEXTSECTION
        Next NextEle
    End If
Next FindEle

NEXTSECTION:

If Counter =1 Then
    Set NEWTrain = NextEle.AddElement("CEM_Tunnel_Trains",
100,100)
    For Each FourthEle In NextEle.ChildElements
        If FourthEle.ElementType="CEM_Tunnel_TrackLayout" Then
            Set FifthEle = FourthEle
        End If
    Next FourthEle
  Set Rel = NEWTrain.CreateRelation
  Set Rel.srcConnection=NEWTrain.ConnectionPoints("Out")
  Set Rel.dstConnection =FifthEle.ConnectionPoints("In1")
  NEWTrain.AddRelation (Rel)
ElseIf Counter >1 Then
    If Elements("1481597")("ResName") <> "Double" Then
```

```
            Elements("1481597")("ResName") = "Double"
                Elements("10")("ShiftLength") = 1200
                Elements("10")("ShiftStart") = 0200
                Elements("10")("CoffeeBreakStart") = 0600
                Elements("10")("LunchBreakStart") = 1000
                Elements("10")("ShiftEnd") = 2200
            End If
        End If
    End If

    F108156 =0
End Function
```

A8.2.10 Element 10 Belief Network Effect Element



Parameter	Value
Belief Network Effect Node	Principle3
State of Concern	Yes
Beleif Value	0.99
Current Belief	0.00
Effect (Linked Value)	LINKED

Figure A8-16. Belief Network Effect Modeling Element Parameters

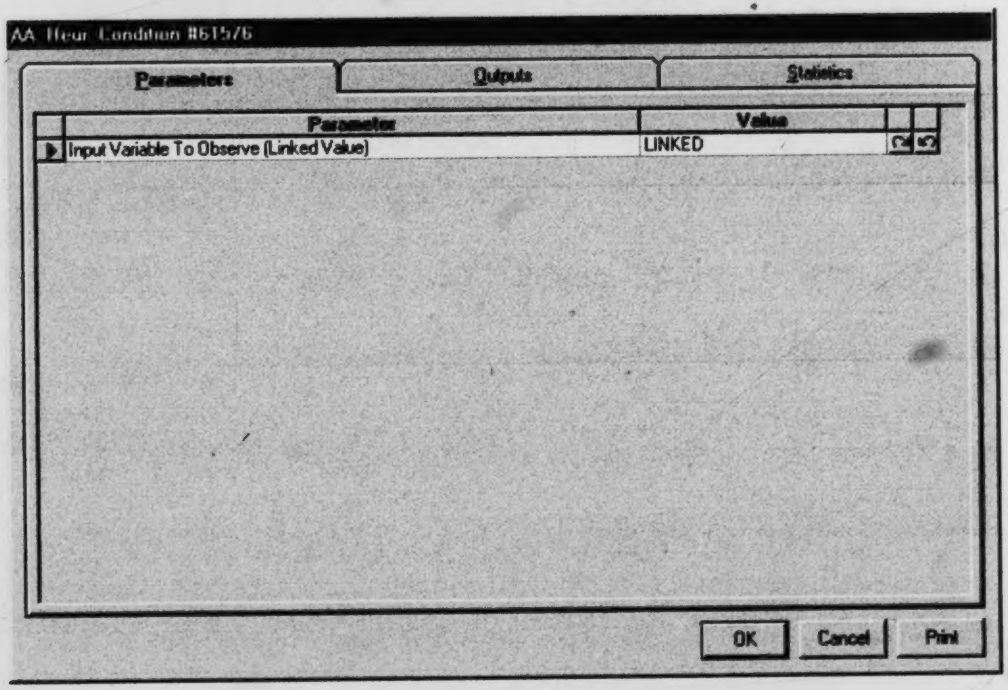
A8.2.10.1 Effect (Linked Value) Linked Code

```
Public Function F177673(ob As CFCSim_ModelingElementInstance)

        Dim FindEle As CFCSim_ModelingElementInstance

        Dim NextEle As CFCSim_ModelingElementInstance

        Dim ThirdEle As CFCSim_ModelingElementInstance

        Dim Counter As Integer

        Dim NEWTrain As CFCSim_ModelingElementInstance
```

```
    Dim Rel As CFCSim_Relation

    Dim FourthEle As CFCSim_ModelingElementInstance

    Dim FifthEle As CFCSim_ModelingElementInstance


    If SimTime > 10000 Then


        For Each FindEle In ob.Parent.Parent.ChildElements

            If FindEle.ElementType = "CEM_Tunnel_Root" Then

                For Each NextEle In FindEle.ChildElements

                    Counter = 0

                    If  NextEle.ElementType  =  "CEM_Tunnel_Undercut"

Then

                        For Each ThirdEle In NextEle.ChildElements

                            If

ThirdEle.ElementType="CEM_Tunnel_Trains" Then Counter = Counter + 1

                        Next ThirdEle

                    End If

                    If Counter >0 Then GoTo NEXTSECTION

                Next NextEle

            End If

        Next FindEle


        NEXTSECTION:
```

```
        If Counter =1 Then
            Set NEWTrain = NextEle.AddElement("CEM_Tunnel_Trains", 100,100)
            For Each FourthEle In NextEle.ChildElements
                If FourthEle.ElementType="CEM_Tunnel_TrackLayout" Then
                    Set FifthEle = FourthEle
                End If
            Next FourthEle
          Set Rel = NEWTrain.CreateRelation
          Set Rel.srcConnection=NEWTrain.ConnectionPoints("Out")
          Set Rel.dstConnection =FifthEle.ConnectionPoints("In1")
          NEWTrain.AddRelation (Rel)
        End If
    End If
    F177673 =0
End Function
```

A8.2.11 Element 11 Rule Based Condition Element


AA Heur Condition #61576
Parameters
Outputs
Statistics
OK
Cancel
Print

	Parameter	Value
▶	Input Variable To Observe (Linked Value)	LINKED

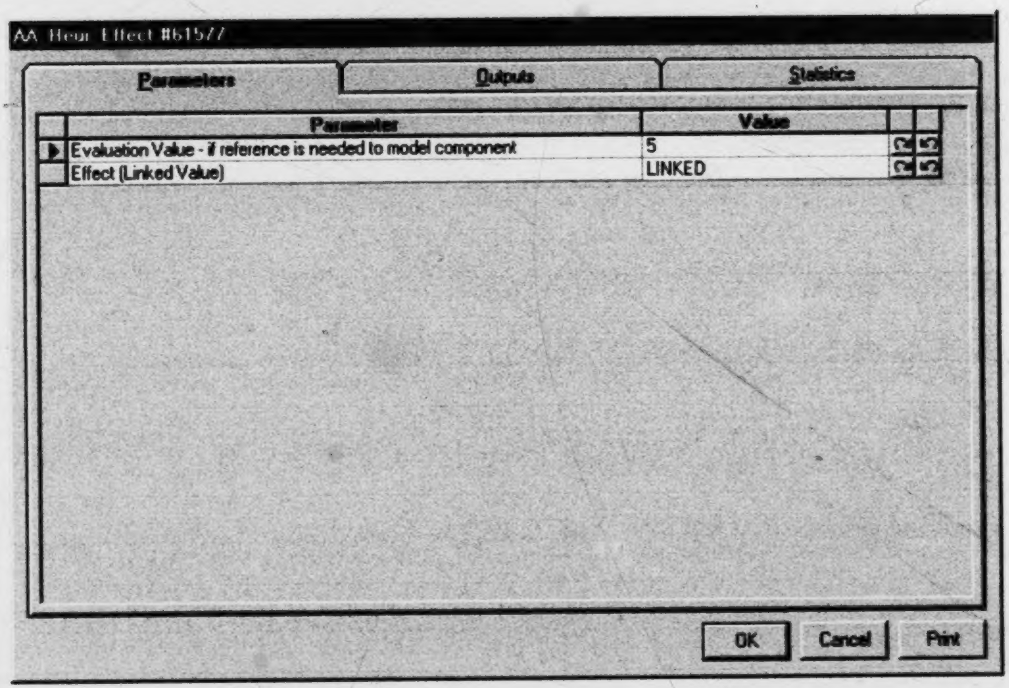
Figure A8-17. Rule Based Condition Modeling Element Parameters

A8.2.11.1 Set Condition To Define State (Linked Value) Linked Code

```
Public Function F61578(ob As CFCSim_ModelingElementInstance)

        Dim Chainage As Double

        Dim Invert As Double

        Dim SoilType As Integer


        Invert = Elements("8")("FinishedLength") * 0.0037669 + 662.708
```

```
    If Invert <=Elements("175")("Depth9") Then
            SoilType = Elements("175")("Soil9")
    ElseIf Invert <=Elements("175")("Depth8") Then
            SoilType = Elements("175")("Soil8")
    ElseIf Invert <=Elements("175")("Depth7") Then
            SoilType = Elements("175")("Soil7")
    ElseIf Invert <=Elements("175")("Depth6") Then
            SoilType = Elements("175")("Soil6")
    ElseIf Invert <=Elements("175")("Depth5") Then
            SoilType = Elements("175")("Soil5")
    ElseIf Invert <=Elements("175")("Depth4") Then
            SoilType = Elements("175")("Soil4")
    ElseIf Invert <=Elements("175")("Depth3") Then
            SoilType = Elements("175")("Soil3")
    ElseIf Invert <=Elements("175")("Depth2") Then
            SoilType = Elements("175")("Soil2")
    Else
            SoilType = Elements("175")("Soil1")
    End If
        Elements("168")("Evaluation")=SoilType

        F61578 = SoilType
End Function
```

A8.2.12 Element 12 Rule Based Effect Element


AA Heur Effect #61577
Parameters
Outputs
Statistics
OK
Cancel
Print

	Parameter	Value
▶	Evaluation Value - if reference is needed to model component	5
	Effect (Linked Value)	LINKED

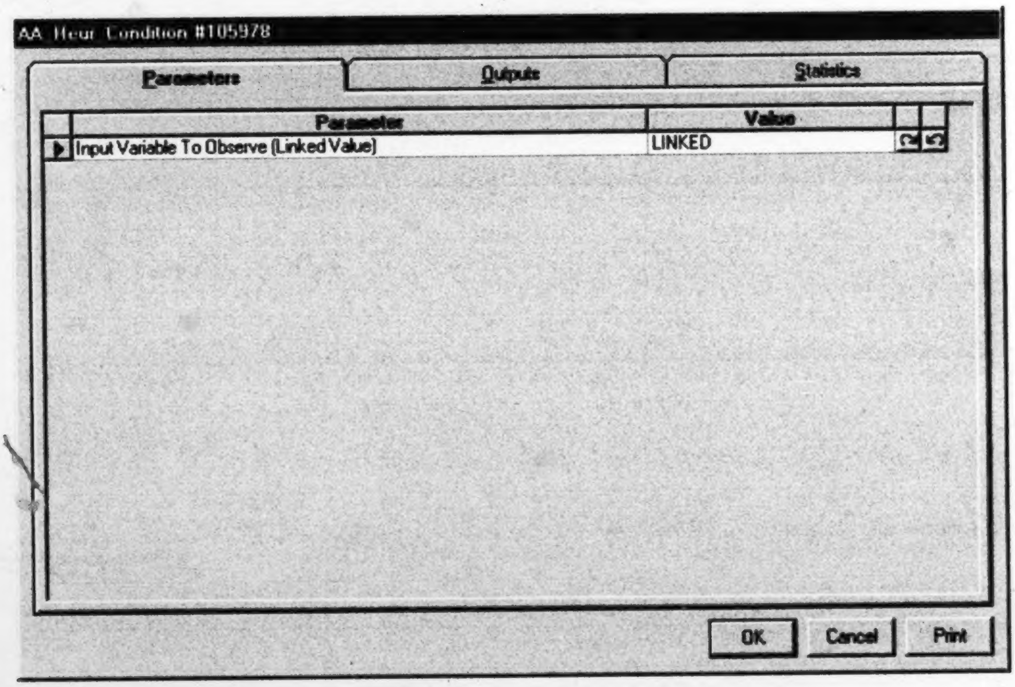
Figure A8-18. Rule Based Effect Modeling Element Parameters

A8.2.12.1 Effect (Linked Value) Linked Code

```
Public Function F61592(ob As CFCSim_ModelingElementInstance)


        Dim SoilType As Integer


        SoilType =ob("Evaluation")
        If SoilType = 2 Then
                Elements("8")("AdvanceRate").Distribution.DistType = CFC_Beta
        Elements("8")("AdvanceRate").Distribution.ParameterValue(0) = 1.6321
        Elements("8")("AdvanceRate").Distribution.ParameterValue(1) = 1.3121
        Elements("8")("AdvanceRate").Distribution.ParameterValue(2) = 1.209
        Elements("8")("AdvanceRate").Distribution.ParameterValue(3) = 5.625
ElseIf SoilType = 5 Then
        Elements("8")("AdvanceRate").Distribution.DistType = CFC_Beta
        Elements("8")("AdvanceRate").Distribution.ParameterValue(0) = 3.481
        Elements("8")("AdvanceRate").Distribution.ParameterValue(1) = 2.9
        Elements("8")("AdvanceRate").Distribution.ParameterValue(2) = 0.995
        Elements("8")("AdvanceRate").Distribution.ParameterValue(3) = 8.077
ElseIf SoilType = 8 Then
        Elements("8")("AdvanceRate").Distribution.DistType = CFC_Beta
        Elements("8")("AdvanceRate").Distribution.ParameterValue(0) = 1.96
        Elements("8")("AdvanceRate").Distribution.ParameterValue(1) = 2.01
        Elements("8")("AdvanceRate").Distribution.ParameterValue(2) = 2.72
        Elements("8")("AdvanceRate").Distribution.ParameterValue(3) = 9
ElseIf SoilType = 9 Then
        Elements("8")("AdvanceRate").Distribution.DistType = CFC_Triangular
        Elements("8")("AdvanceRate").Distribution.ParameterValue(0) = 0.725
        Elements("8")("AdvanceRate").Distribution.ParameterValue(1) = 5.392
        Elements("8")("AdvanceRate").Distribution.ParameterValue(2) = 7.945
Else
        Elements("8")("AdvanceRate").Distribution.DistType = CFC_Triangular
```

```
            Elements("8")("AdvanceRate").Distribution.ParameterValue(0) = 2

            Elements("8")("AdvanceRate").Distribution.ParameterValue(1) = 5

            Elements("8")("AdvanceRate").Distribution.ParameterValue(2) = 7

        End If


    F61592 =0

End Function
```

A8.2.13 Element 13 Rule Based Condition Element


AA Heur Condition #105978
Parameters
Outputs
Statistics
OK
Cancel
Print

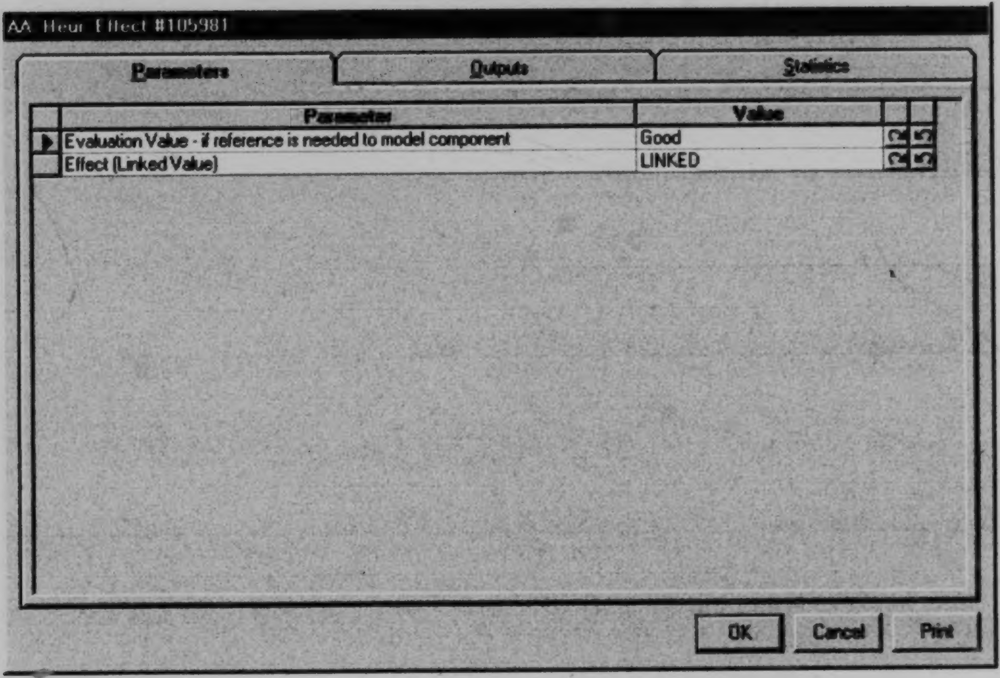
Parameter	Value
Input Variable To Observe (Linked Value)	LINKED

Figure A8-19. Rule Based Condition Modeling Element Parameters

A8.2.13.1 Set Condition To Define State (Linked Value) Linked Code

```
Public Function F105979(ob As CFCSim_ModelingElementInstance)

        Dim ProjectDays As Integer
        Dim DaysWorked As Integer
        Dim DaysLeft As Integer
        Dim AvailableShifts As Integer
        Dim BoredLength As Double
        Dim UnboredLength As Double
        Dim CurrentProductivity As Double
        Dim ShiftsRequired As Double

        If SimTime > 5000 Then
                ProjectDays = 181
                DaysWorked = SimTime/Elements("100")("ShiftLength")
                DaysLeft = ProjectDays-DaysWorked
                AvailableShifts = DaysLeft*2
                BoredLength = Elements("8")("FinishedLength")
                UnboredLength = Elements("8")("Length")-BoredLength
                CurrentProductivity = BoredLength/DaysWorked
                ShiftsRequired = UnboredLength/CurrentProductivity

                If ShiftsRequired >= AvailableShifts Then
                Elements("170")("Evaluation") = "SecondShiftNeeded"

            Else

                Elements("170")("Evaluation") = "Good"

            End If

        Else

            Elements("170")("Evaluation") = "Good"

        End If


        F105979 = 0


End Function
```

A8.2.14 Element 14 Rule Based Effect Element


AA Heur Effect #105981
Parameters
Outputs
Statistics
OK
Cancel
Print

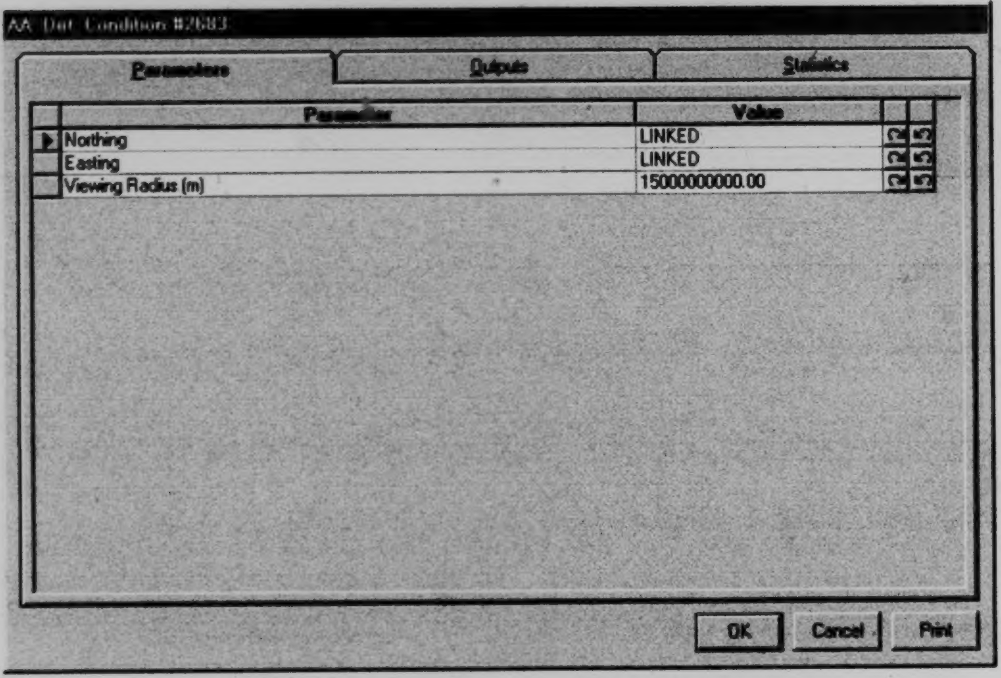
Parameter	Value
Evaluation Value - if reference is needed to model component	Good
Effect (Linked Value)	LINKED

Figure A8-20. Rule Based Effect Modeling Element Parameters

A8.2.14.1 Effect (Linked Value) Linked Code

```
Public Function F105984(ob As CFCSim_ModelingElementInstance)


        If SimTime < 500 Then        ob("Evaluation") = "Good"

    F105984 = 0


End Function
```

A8.2.15 Element 15 Algorithm Condition Element


AA Dat Condition #2683
Parameters
Outputs
Statistics
OK
Cancel
Print

Parameter	Value
Northing	LINKED
Easting	LINKED
Viewing Radius (m)	15000000000.00

Figure A8-21. Algorithm Condition Modeling Element Parameters

A8.2.15.1 Northing Linked Code

```
Public Function F2699(ob As CFCSim_ModelingElementInstance)

        Dim Chainage As Double

        Dim Northing As Double


        Chainage = Elements("8")("FinishedLength")
```

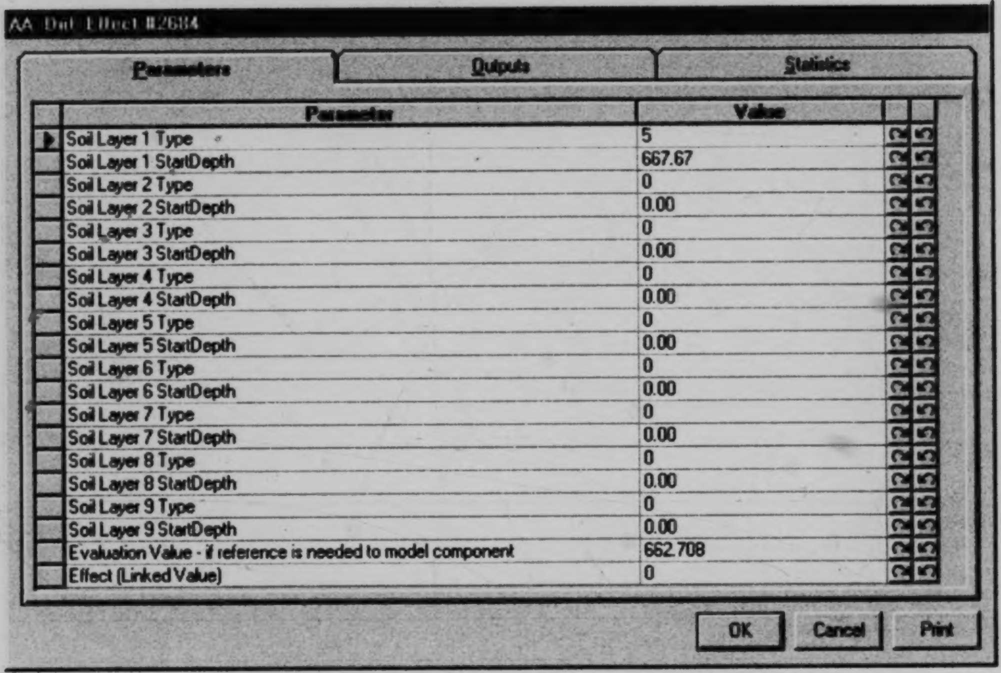
```
Elements("2684")("Evaluation") =662.708 + Chainage * 0.0037669 '        Invert Elevation

If Chainage >=0 Then
    If Chainage < 89.440 Then
        Chainage = Chainage
        Northing = 5943642.8720 - Chainage * Sin(0.400)
    End If

ElseIf Chainage >=89.440 Then
    If Chainage < 379.110 Then
        Chainage = Chainage-89.440
        Northing = 5943642.248+ Chainage * Sin(13.431)
    End If

ElseIf Chainage >=379.110 Then
    If Chainage < 695.709 Then
        Chainage = Chainage-379.110
        Northing = 5943708.878+ Chainage * Sin(27.262)
    End If

ElseIf Chainage >=695.709 Then
    If Chainage <908.028 Then
```

```
                    Chainage = Chainage-695.709

                    Northing = 5943853.897+ Chainage * Sin(15.229)

          End If


ElseIf Chainage >=908.028 Then

          If Chainage <1426.145 Then

                    Chainage = Chainage-908.028

                    Northing = 5943909.259+ Chainage * Sin(3.196)

          End If


ElseIf Chainage >=1426.145 Then

          If Chainage <1517.633 Then

                    Chainage = Chainage-1426.145

                    Northing = 5943938.147+ Chainage * Sin(5.817)

          End If


ElseIf Chainage >=1517.633 Then

                    Chainage = Chainage-1517.633

                    Northing = 5943947.417+ Chainage * Sin(8.438)

End If


F2699 =Northing

End Function
```

A8.2.15.2 Easting Linked Code

```
Public Function F2700(ob As CFCSim_ModelingElementInstance)
        Dim Chainage As Double
        Dim Easting As Double

        Chainage = Elements("8")("FinishedLength")

        If Chainage >=0 Then
                If Chainage < 89.440 Then
                        Chainage = Chainage
                        Easting = 31887.8510 - Chainage * Cos(0.400)
                End If

        ElseIf Chainage >=89.440 Then
                If Chainage < 379.110 Then
                        Chainage = Chainage-89.440
                        Easting = 31798.413 - Chainage * Cos(13.431)
                End If

        ElseIf Chainage >=379.110 Then
                If Chainage < 695.709 Then
                        Chainage = Chainage-379.110
                        Easting = 31519.405 - Chainage * Cos(27.262)
        End If


ElseIf Chainage >=695.709 Then

        If Chainage <908.028 Then

                Chainage = Chainage-695.709

                Easting = 31237.972 - Chainage * Cos(15.229)

        End If


ElseIf Chainage >=908.028 Then

        If Chainage <1426.145 Then

                Chainage = Chainage-908.028

                Easting = 31034.611 - Chainage * Cos(3.196)

        End If


ElseIf Chainage >=1426.145 Then

        If Chainage <1517.633 Then

                Chainage = Chainage-1426.145

                Easting = 30517.300 - Chainage * Cos(5.817)

        End If


ElseIf Chainage >=1517.633 Then

                Chainage = Chainage-1517.633

                Easting = 30426.314 - Chainage * Cos(8.438)
        End If
   F2700 =Easting
End Function
```

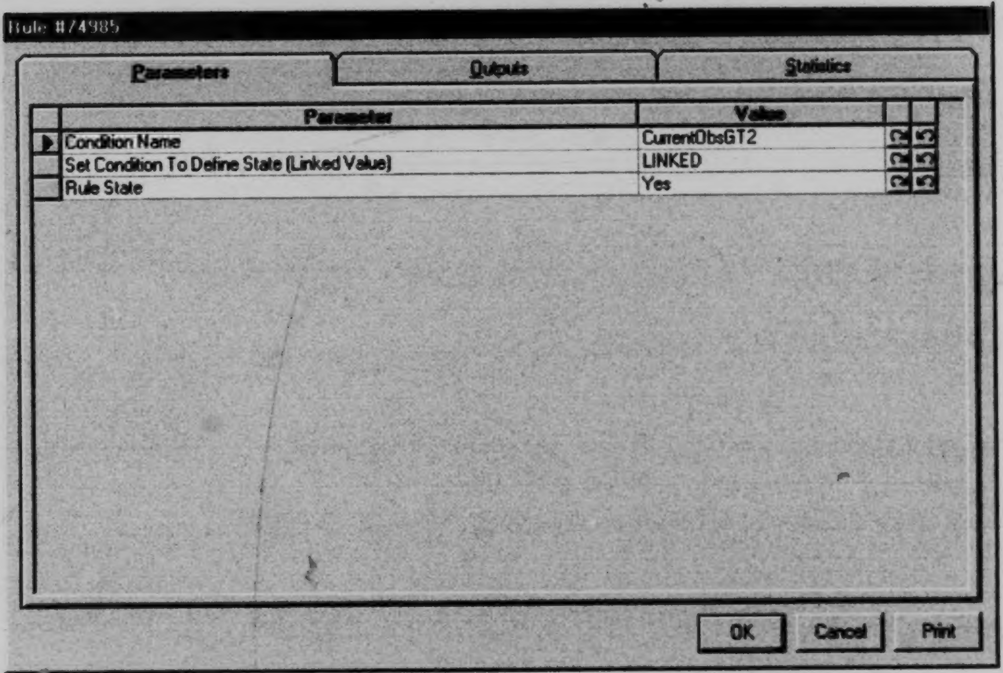
A8.2.16 Element 16 Algorithm Effect Element



Parameter	Value
Soil Layer 1 Type	5
Soil Layer 1 StartDepth	667.67
Soil Layer 2 Type	0
Soil Layer 2 StartDepth	0.00
Soil Layer 3 Type	0
Soil Layer 3 StartDepth	0.00
Soil Layer 4 Type	0
Soil Layer 4 StartDepth	0.00
Soil Layer 5 Type	0
Soil Layer 5 StartDepth	0.00
Soil Layer 6 Type	0
Soil Layer 6 StartDepth	0.00
Soil Layer 7 Type	0
Soil Layer 7 StartDepth	0.00
Soil Layer 8 Type	0
Soil Layer 8 StartDepth	0.00
Soil Layer 9 Type	0
Soil Layer 9 StartDepth	0.00
Evaluation Value - if reference is needed to model component	662.708
Effect (Linked Value)	0

Figure A8-22. Algorithm Effect Modeling Element Parameters

A8.2.17 Element 17 Belief Network Condition Element


Rule #74985
Parameters
Outputs
Statistics
OK
Cancel
Print

Parameter	Value
Condition Name	CurrentObsGT2
Set Condition To Define State (Linked Value)	LINKED
Rule State	Yes

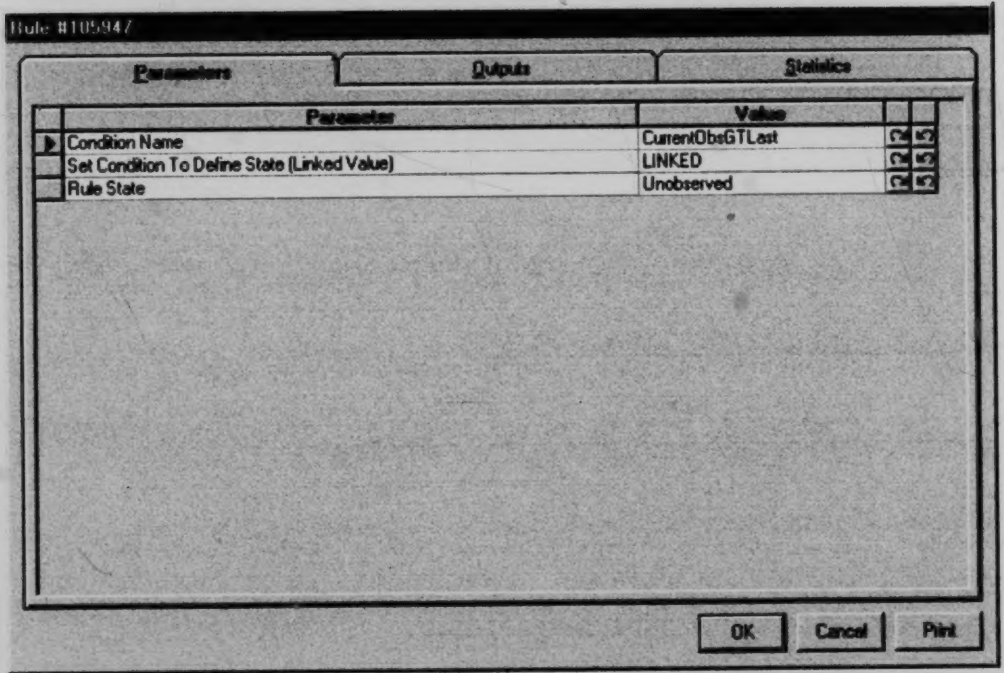
Figure A8-23. Belief Network Condition Modeling Element Parameters

A8.2.17.1 Set Condition To Define State (Linked Value) Linked Code

```
Public Function F101242(ob As CFCSim_ModelingElementInstance)

    Dim WaitTime As Double

    If Elements("97")("Total") > Elements("98")("Total") Then
        WaitTime = Elements("97")("Total")-Elements("98")("Total")
        If WaitTime>30 Then
```

```
                        ob("RuleState") = "Yes"
                ElseIf WaitTime<30 Then
                        ob("RuleState") = "No"
                Else
                        ob("RuleState") = "Unobserved"
                End If
        Else
                ob("RuleState") = "Unobserved"
        End If

    F101242 = 0
End Function
```

A8.2.18 Element 18 Belief Network Condition Element


Rule #105947
Parameters
Outputs
Statistics
OK
Cancel
Print

Parameter	Value
Condition Name	CurrentObsGTLast
Set Condition To Define State (Linked Value)	LINKED
Rule State	Unobserved

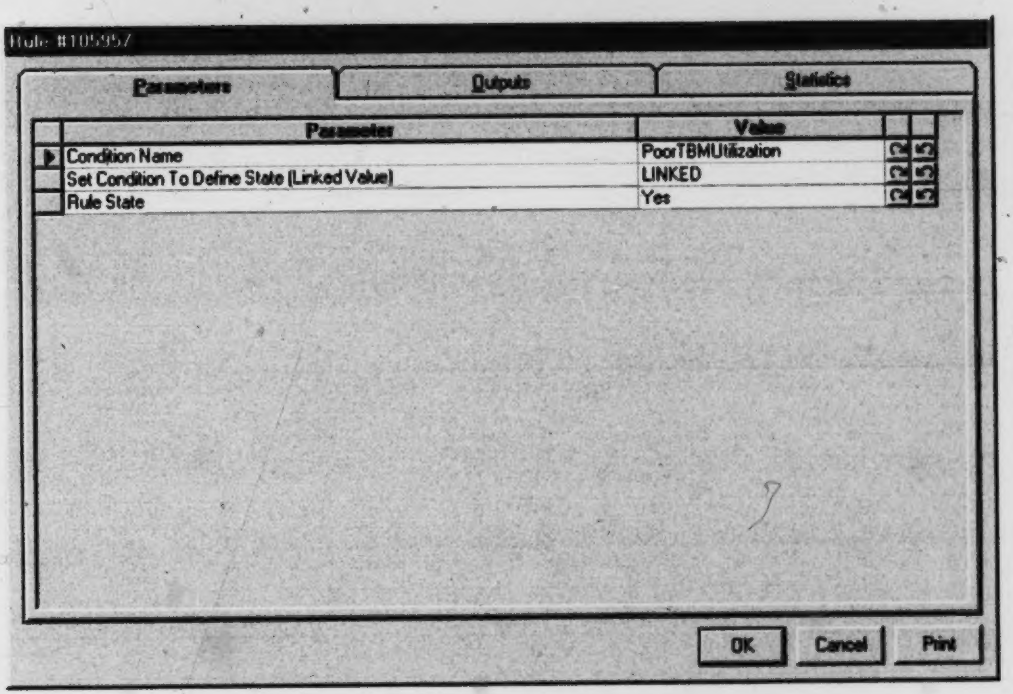
Figure A8-24. Belief Network Condition Modeling Element Parameters

A8.2.18.1 Set Condition To Define State (Linked Value) Linked Code

```
Public Function F105948(ob As CFCSim_ModelingElementInstance)


        Dim WaitTime As Double


        If Elements("97")("Total") > Elements("98")("Total") Then

                WaitTime = Elements("97")("Total")-Elements("98")("Total")

                If SimTime > 500 Then
```

```
                    If WaitTime>Elements("170")("Total") Then
                              ob("RuleState") = "Yes"
                    ElseIf WaitTime<Elements("170")("Total") Then
                              ob("RuleState") = "No"
                    Else
                              ob("RuleState") = "Unobserved"
                    End If
          End If
          Elements("170")("Total") =WaitTime
Else
          ob("RuleState") = "Unobserved"
End If


End Function
```

A8.2.19 Element 19 Belief Network Condition Element


Rule #105957
Parameters
Outputs
Statistics
OK
Cancel
Print

Parameter	Value
Condition Name	PoorTBMUtilization
Set Condition To Define State (Linked Value)	LINKED
Rule State	Yes

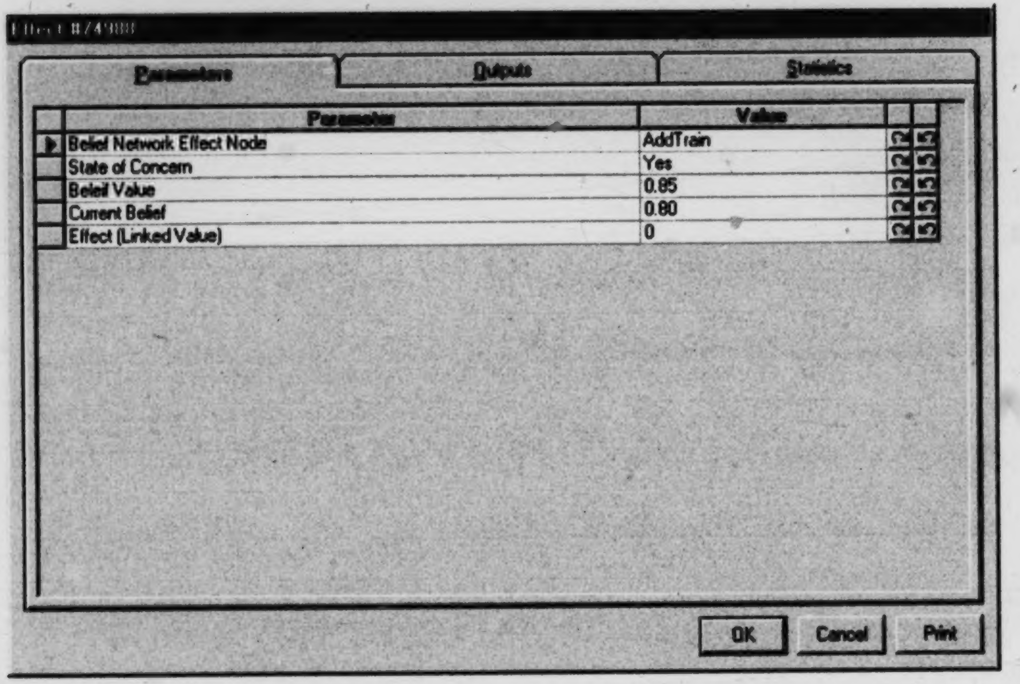
Figure A8-25. Belief Network Condition Modeling Element Parameters

A8.2.19.1 Set Condition To Define State (Linked Value) Linked Code

```
Public Function F105958(ob As CFCSim_ModelingElementInstance)

    If Elements("16").stat("Utilization").GlobalAverage <50 Then
        ob("RuleState") = "Yes"
    ElseIf Elements("16").stat("Utilization").GlobalAverage >= 50 Then
        ob("RuleState") = "No"
    Else
```

```
            ob("RuleState") = "Unobserved"
        End If
    F105958=0
End Function
```

A8.2.20 Element 20 Belief Network Effect Element


Parameters
Outputs
Statistics
OK
Cancel
Print

Parameter	Value
Belief Network Effect Node	AddTrain
State of Concern	Yes
Belief Value	0.85
Current Belief	0.80
Effect (Linked Value)	0

Figure A8-26. Belief Network Effect Modeling Element Parameters

APPENDIX 9 – Agent Model Element Information

A9.1 Belief Network Agent

The following subsections provide the information pertaining to the construction of the modeling elements that make up the belief network agent presented in this research. In addition to the programming code presented a reference was made within the Simphony editing environment to the Dynamic Library Links (DLL's) provided with the Microsoft Research's Bayesian Network Authoring and Evaluation Tool MSBNx (1.4). Another reference was also made to the Microsoft Scripting Runtime DLL that was used to manage how file types were presented to the agents so that they could be filtered and placed in the parent agent element in a list box format. It should also be noted that when developing the belief network agent the variables in the belief networks were divided into input and output variables. This was done following work carried out by Van Tol (2000). While the agent presented in the following subsections follows this division of belief network variables into input and output variables, in retrospect this division is not required.

A9.1.1 Parent Agent Element Programming Code

```
Option Explicit
```

```
Public Function AA_OnCreate(ob As CFCSim_ModelingElementInstance, x As Single, y As
Single) As Boolean
        AA_OnCreate=True
        ob.OnCreate x,y,True
```

```
	ob.AddAttribute "BeliefFile","Belief Network
File",CFC_Text,CFC_ListBox,CFC_ReadWrite
	ob.AddAttribute "FirstObs","Time of First
Observation",CFC_Numeric,CFC_Single,CFC_ReadWrite
	ob.AddAttribute "ObservationInterval","Observation
Interval",CFC_Numeric,CFC_Single,CFC_ReadWrite

	ob.Attr("BeliefFile") = "Train.xbn"
	ob.Attr("ObservationInterval") = 480
	ob.Attr("FirstObs") = 480

	ob.SetNumCoordinates 2
	ob.CoordinatesX(0)=x
	ob.CoordinatesY(0)=y
	ob.CoordinatesX(1)=x+70
	ob.CoordinatesY(1)=y+49
End Function
```

```
Public Sub AA_OnDragDraw(ob As CFCSim_ModelingElementInstance)
	ob.OnDraw
End Sub
```

```
Public Sub AA_OnDraw(ob As CFCSim_ModelingElementInstance)
```

```
    CDC.RenderPicture
"harrys1.bmp",ob.CoordinatesX(0),ob.CoordinatesY(0),ob.CoordinatesX(1)-
ob.CoordinatesX(0),ob.CoordinatesY(1)-ob.CoordinatesY(0)

    If ob.Selected Then
        CDC.ChangeLineStyle CFC_DOT,1,RGB(255,0,0)
        CDC.Rectangle                    ob.CoordinatesX(0)-2,ob.CoordinatesY(0)-
2,ob.CoordinatesX(1)+2,ob.CoordinatesY(1)+2
    End If

    CDC.ChangeLineStyle CFC_SOLID,1,RGB(0,0,0)
End Sub
```

```
Public Sub AA_OnListBoxInitialize(ob As CFCSim_ModelingElementInstance, attr As
CFCSim_Attribute, lstList As Object)
    Dim MyFolder As New FileSystemObject
    Dim BNFolder As Folder
    Dim MyFile As File

    Select Case attr.Name
    Case "BeliefFile"
        Set BNFolder = MyFolder.GetFolder("C:\BNF\")
```

```
                    For Each MyFile In BNFolder.Files

                    If MyFile.Type = "MSBNX Bayesian Model" Then

                    lstList.additem MyFile.Name

                    End If

                    Next MyFile

          End Select

End Sub
```

```
Public Sub AA_OnSimulationInitialize(ob As CFCSim_ModelingElementInstance)

          ob.AddEvent "CheckRules",True

          ob.AddEvent "CheckBN"

End Sub
```

```
Public   Sub   AA_OnSimulationInitializeRun(ob   As   CFCSim_ModelingElementInstance,

RunNum As Integer)

          Dim ent As CFCSim_Entity

          Set ent = ob.AddEntity


          ob.ScheduleEvent ent,"CheckRules",ob("FirstObs")


End Sub
```

```
Public Sub AA_OnSimulationProcessEvent(ob As CFCSim_ModelingElementInstance,
MyEvent As String, Entity As CFCSim_Entity)
        Dim Rule As CFCSim_ModelingElementInstance
        Dim CurrRuleState As String
        Dim EState As Variant
        Dim ERule As Variant

        Select Case MyEvent
        Case "CheckRules"
                'get each rule to evaluate itself
                For Each Rule In ob.ChildElements
                        If Rule.ElementType ="Rule" Then
                                Dim ent2 As CFCSim_Entity
                                Set ent2 = ob.AddEntity
                                Rule.ScheduleEvent ent2,"CheckRule",0
                        End If
                Next

                Dim ent3 As CFCSim_Entity
                Set ent3 = ob.AddEntity
                Dim ent4 As CFCSim_Entity
                Set ent4 = ob.AddEntity
        ob.ScheduleEvent ent3,"CheckRules", ob("ObservationInterval")
        ob.ScheduleEvent ent4,"CheckBN", 0

    Case"CheckBN"

        'send info to net and see if value is good or not - if good run effect
        Dim aMsbn As New MSBN3Lib.MSBN
        Dim MyModel As MSBN3Lib.Model
        Dim Infer As MSBN3Lib.Engine
        Dim Node As MSBN3Lib.Node

        Set   MyModel   =   aMsbn.Models.Add("Auto",   FileName:="C:\BNF\"   &
ob("BeliefFile"))
        Set Infer = MyModel.Engine

        For Each Rule In ob.ChildElements
            If Rule.ElementType ="Rule" Then
                CurrRuleState = Rule("RuleState")
                For Each Node In MyModel.ModelNodes
                    If Rule("RuleName") = Node.Name Then
                        If Rule("RuleState") <> "Unobserved" Then
                            ERule = Rule("RuleState")
                            Infer.Evidence.Add Node.Name, ERule
                        End If
                    End If
                End If
            Next Node
        End If
    Next Rule

    For Each Rule In ob.ChildElements
        If Rule.ElementType ="Effect" Then
            For Each Node In MyModel.ModelNodes
                If Rule("EffectName") = Node.Name Then
                    If Rule("EffectState") <> "Unobserved" Then
                        EState = Rule("EffectState")
                        Rule("CurrentBelief")                 =
Infer.Belief(Node.Name, EState)
                        Dim ent5 As CFCSim_Entity
                        Set ent5 = ob.AddEntity

    Rule.ScheduleEvent(ent5,"ProcessEffect", 0)
                    End If
                End If
            Next Node
        End If
    Next Rule
```

```
        End Select
End Sub
```

A9.1.2 Condition Element Programming Code

```
Option Explicit
```

```
Public Function Rule_OnCreate(ob As CFCSim_ModelingElementInstance, X As Single, y
As Single) As Boolean
        Rule_OnCreate=True
        ob.OnCreate X,y,True

        ob.AddAttribute "RuleName","Condition
Name",CFC_Text,CFC_ListBox,CFC_ReadWrite
        ob.AddAttribute "Condition","Set Condition To Define State (Linked
Value)",CFC_Text,CFC_Single,CFC_ReadWrite
        ob.AddAttribute "RuleState","Rule State",CFC_Text,CFC_ListBox,CFC_ReadWrite

        ob.Attr("RuleState") = "Unobserved"

        ob.SetNumCoordinates 2
        ob.CoordinatesX(0)=X
        ob.CoordinatesY(0)=y
        ob.CoordinatesX(1)=X+80

        ob.CoordinatesY(1)=y+80

End Function
```

```
Public Sub Rule_OnDragDraw(ob As CFCSim_ModelingElementInstance)

        ob.OnDraw

End Sub
```

```
Public Sub Rule_OnDraw(ob As CFCSim_ModelingElementInstance)

        CDC.RenderPicture

"if.bmp",ob.CoordinatesX(0),ob.CoordinatesY(0),ob.CoordinatesX(1)-

ob.CoordinatesX(0),ob.CoordinatesY(1)-ob.CoordinatesY(0)


        If ob.Selected Then

                CDC.ChangeLineStyle CFC_DOT,1,RGB(255,0,0)

                CDC.Rectangle                   ob.CoordinatesX(0)-2,ob.CoordinatesY(0)-

2,ob.CoordinatesX(1)+2,ob.CoordinatesY(1)+2

        End If


        CDC.ChangeLineStyle CFC_SOLID,1,RGB(0,0,0)

End Sub
```

```
Public Sub Rule_OnListBoxInitialize(ob As CFCSim_ModelingElementInstance, attr As
CFCSim_Attribute, lstList As Object)
        Dim aMSBN As New MSBN3Lib.MSBN
        Dim BNModel As MSBN3Lib.Model
        Dim Node As MSBN3Lib.Node
        Dim Nstate As MSBN3Lib.state

        Select Case attr.Name
        Case "RuleName"

                Set BNModel = aMSBN.Models.Add("Auto", FileName:="C:\BNF\" &
ob.Parent.Attr("BeliefFile"))

                For Each Node In BNModel.ModelNodes
                If Node.Description = "Input" Then
                                lstList.additem Node.Name
                End If
                Next Node

        Case "RuleState"

                lstList.additem "Unobserved"
        Set BNModel = aMSBN.Models.Add("Auto", FileName:="C:\BNF\" & ob.Parent.Attr("BeliefFile"))

        For Each Node In BNModel.ModelNodes
          If Node.Description = "Input" Then
                If Node.Name = ob.Attr("RuleName") Then
                    For Each Nstate In Node.States
                        lstList.additem Nstate.Name
               Next Nstate
                End If
            End If
        Next Node
    End Select
End Sub
```

```
Public Sub Rule_OnListBoxSelectItem(ob As CFCSim_ModelingElementInstance, attr As CFCSim_Attribute, lstList As Object)
    attr.Value = lstList.text
    ob(attr.Name) = attr.Value
End Sub
```

```
Public Sub Rule_OnSimulationInitialize(ob As CFCSim_ModelingElementInstance)
    ob.AddEvent "CheckRule",True
```

```
End Sub
```

```
Public Sub Rule_OnSimulationProcessEvent(ob As CFCSim_ModelingElementInstance,
MyEvent As String, Entity As CFCSim_Entity)
Dim X As String
    Select Case MyEvent
        Case "CheckRule"
                X = ob("Condition")
    End Select
End Sub
```

A9.1.3 Effect Element Programming Code

```
Option Explicit
```

```
Public Function Effect_OnCreate(ob As CFCSim_ModelingElementInstance, X As Single, y
As Single) As Boolean
    Effect_OnCreate=True
    ob.OnCreate X,y,True

    ob.AddAttribute "EffectName","Belief Network Effect
Node",CFC_Text,CFC_ListBox,CFC_ReadWrite
```

```
    ob.AddAttribute "EffectState","State of Concern",CFC_Text,CFC_ListBox,CFC_ReadWrite
    ob.AddAttribute "BeliefValue", "Beleif Value",CFC_Numeric,CFC_Single,CFC_ReadWrite
    ob.AddAttribute "CurrentBelief","Current Belief", CFC_Numeric,CFC_Single,CFC_ReadWrite
    ob.AddAttribute "Effect","Effect (Linked Value)",CFC_Text,CFC_Single,CFC_ReadWrite

    ob("BeliefValue") = 0.75
    ob("CurrentBelief") = 0

    ob.SetNumCoordinates 2
    ob.CoordinatesX(0)=X
    ob.CoordinatesY(0)=y
    ob.CoordinatesX(1)=X+80
    ob.CoordinatesY(1)=y+80
End Function
```

```
Public Sub Effect_OnDragDraw(ob As CFCSim_ModelingElementInstance)
    ob.OnDraw
End Sub
```

```
Public Sub Effect_OnDraw(ob As CFCSim_ModelingElementInstance)
        CDC.RenderPicture
"then.bmp",ob.CoordinatesX(0),ob.CoordinatesY(0),ob.CoordinatesX(1)-
ob.CoordinatesX(0),ob.CoordinatesY(1)-ob.CoordinatesY(0)

        If ob.Selected Then
                CDC.ChangeLineStyle CFC_DOT,1,RGB(255,0,0)
                CDC.Rectangle                    ob.CoordinatesX(0)-2,ob.CoordinatesY(0)-
2,ob.CoordinatesX(1)+2,ob.CoordinatesY(1)+2
        End If

        CDC.ChangeLineStyle CFC_SOLID,1,RGB(0,0,0)
End Sub
```

```
Public Sub Effect_OnListBoxInitialize(ob As CFCSim_ModelingElementInstance, attr As
CFCSim_Attribute, lstList As Object)
        Dim aMSBN As New MSBN3Lib.MSBN
        Dim BNModel As MSBN3Lib.Model
        Dim Node As MSBN3Lib.Node
        Dim Nstate As MSBN3Lib.state

        Select Case attr.Name
        Case "EffectName"
```

```
            Set  BNModel  =  aMSBN.Models.Add("Auto",  FileName:="C:\BNF\"  &
ob.Parent.Attr("BeliefFile"))

            For Each Node In BNModel.ModelNodes
            If Node.Description = "Output" Then
                        lstList.additem Node.Name
            End If
            Next Node

      Case "EffectState"

            Set  BNModel  =  aMSBN.Models.Add("Auto",  FileName:="C:\BNF\"  &
ob.Parent.Attr("BeliefFile"))

            For Each Node In BNModel.ModelNodes
              If Node.Description = "Output" Then
                        If Node.Name = ob.Attr("EffectName") Then
                              For Each Nstate In Node.States
                                    lstList.additem Nstate.Name
                        Next Nstate
                          End If
                  End If
```

```
        Next Node
    End Select
End Sub
```

```
Public Sub Effect_OnListBoxSelectItem(ob As CFCSim_ModelingElementInstance, attr As
CFCSim_Attribute, lstList As Object)
    attr.Value = lstList.text
    ob(attr.Name) = attr.Value
End Sub
```

```
Public Sub Effect_OnSimulationInitialize(ob As CFCSim_ModelingElementInstance)
    ob.AddEvent "ProcessEffect",True
End Sub
```

```
Public Sub Effect_OnSimulationProcessEvent(ob As CFCSim_ModelingElementInstance,
MyEvent As String, Entity As CFCSim_Entity)
    Dim X As String

    Select Case MyEvent
    Case "ProcessEffect"
        If ob("CurrentBelief")>=ob("BeliefValue") Then
            X = ob("Effect")
        End If
    End Select
End Sub
```

A9.2 Neural Network Agent

The following subsections provide the information pertaining to the construction of the modeling elements that make up the neural network agent and its observation assistant presented in this research. In developing this agent a DLL was constructed that refer to a Microsoft Visual Basic module (NW32.bas) that references the NeuroWindows neural network engine.

A9.2.1 Parent Agent Element Programming Code

```
Option Explicit
```

```
Public Function AA_ANN_OnCreate(ob As CFCSim_ModelingElementInstance, X As Single, Y As Single) As Boolean
        AA_ANN_OnCreate=True
        ob.OnCreate X,Y,True

        ob.AddAttribute "NeuralFile","Neural Network File",CFC_Text,CFC_ListBox,CFC_ReadWrite
        ob.AddAttribute "FirstObs","Time of First Observation",CFC_Numeric,CFC_Single,CFC_ReadWrite
```

```
    ob.AddAttribute "ObservationInterval","Observation Interval",CFC_Numeric,CFC_Single,CFC_ReadWrite
    ob.AddAttribute "LearnRate","Net Leaning Rate",CFC_Numeric,CFC_Single,CFC_ReadWrite
    ob.AddAttribute "Momentum","Net Training Momentum",CFC_Numeric,CFC_Single,CFC_ReadWrite
    ob.AddAttribute "AverageError","Acceptable Net Training Error (less than 1.0)",CFC_Numeric,CFC_Single,CFC_ReadWrite

    ob.AddAttribute "NumInputs","Number of Input Nodes",CFC_Numeric,CFC_Single,CFC_Hidden
    ob.AddAttribute "NumOutputs","Number of Output Nodes",CFC_Numeric,CFC_Single,CFC_Hidden
    ob.AddAttribute "NetDBID","Network DB ID",CFC_Text,CFC_Single,CFC_Hidden
    ob.AddAttribute "NetNum","NeuroShell Net Number",CFC_Numeric,CFC_Single,CFC_Hidden

    ob.Attr("ObservationInterval") = 10
    ob.Attr("FirstObs") = 0
    ob.Attr("LearnRate") = 0.60
    ob.Attr("Momentum") = 0.90
    ob.Attr("AverageError") = 0.04
```

```
        ob.SetNumCoordinates 2
        ob.CoordinatesX(0)=X
        ob.CoordinatesY(0)=Y
        ob.CoordinatesX(1)=X+70
        ob.CoordinatesY(1)=Y+49

End Function
```

```
Public Sub AA_ANN_OnDragDraw(ob As CFCSim_ModelingElementInstance)
        ob.OnDraw
End Sub
```

```
Public Sub AA_ANN_OnDraw(ob As CFCSim_ModelingElementInstance)
        CDC.RenderPicture
"harrys1.bmp",ob.CoordinatesX(0),ob.CoordinatesY(0),ob.CoordinatesX(1)-
ob.CoordinatesX(0),ob.CoordinatesY(1)-ob.CoordinatesY(0)

        If ob.Selected Then
                CDC.ChangeLineStyle CFC_DOT,1,RGB(255,0,0)
                CDC.Rectangle                    ob.CoordinatesX(0)-2,ob.CoordinatesY(0)-
2,ob.CoordinatesX(1)+2,ob.CoordinatesY(1)+2
        End If
        CDC.ChangeLineStyle CFC_SOLID,1,RGB(0,0,0)
End Sub
```

```
Public Sub AA_ANN_OnListBoxInitialize(ob As CFCSim_ModelingElementInstance, attr
As CFCSim_Attribute, lstList As Object)
        Dim dbs As Database
        Dim Rst As Recordset

        Select Case attr.Name
        Case "NeuralFile"
                Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\Net.mdb")
                Set Rst = dbs.TableDefs("Net").OpenRecordset

                If Rst.RecordCount > 0 Then
                Rst.MoveFirst
                Do Until Rst.EOF
                lstList.additem Rst!NetName
                Rst.MoveNext
                Loop
                End If
        End Select
End Sub
```

```
Public Sub AA_ANN_OnListBoxSelectItem(ob As CFCSim_ModelingElementInstance, attr
As CFCSim_Attribute, lstList As Object)
        Dim dbs As Database
        Dim Rst As Recordset
        Dim Child As CFCSim_ModelingElementInstance
        Dim X As Integer
        Dim Table As TableDef

        attr.Value = lstList.text
        ob(attr.Name) = attr.Value
        Select Case attr.Name
        Case "NeuralFile"
                'clean out db if old table exists
                Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\net.mdb")
                For Each Table In dbs.TableDefs
                        If Table.Name = "ChildInfo" Then dbs.Execute "Drop Table
ChildInfo"
                Next Table
                dbs.Close

                Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\Net.mdb")
                Set Rst = dbs.TableDefs("Net").OpenRecordset
```

```
'get the net info
If Rst.RecordCount > 0 Then
        Rst.MoveFirst
        Do Until Rst.EOF
                If ob("NeuralFile").Value = Rst!NetName Then
                        ob.Attr("NetDBID").Value = Rst!NetId
                        ob.Attr("NumInputs").Value = Rst!InputNum
                        ob.Attr("NumOutputs").Value = Rst!OutputNum
                End If
                Rst.MoveNext
Loop
End If
'clean out the child elements if a new net is selected
If ob.ChildElements.Count> 0 Then
        For Each Child In ob.ChildElements
                Child.Delete
        Next Child
End If
'put in the elements
Dim NewEle As CFCSim_ModelingElementInstance

For X = 1 To ob("NumInputs")
        Set NewEle = ob.AddElement("Condition_ANN",X*100, 50)
```

```
    NewEle("DataType") = "Input" & X
Next X


For X = 1 To ob("NumOutputs")
    Set NewEle = ob.AddElement("Effect_ANN",X*100, 250)
    NewEle("DataType") = "Output" & X
Next X


'feed the info into the new elements
dbs.Execute"Select NetDataInfo.DataType, NetDataInfo.TypeName INTO
ChildInfo FROM Net INNER Join NetDataInfo On Net.NetID = NetDataInfo.NetID
WHERE (Net.NetID)=" & ob("NetDBID")
dbs.Close


Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\Net.mdb")
Set Rst = dbs.TableDefs("ChildInfo").OpenRecordset


For Each Child In ob.ChildElements
    Rst.MoveFirst
    Do Until Rst.EOF

        If Child("DataType") = Rst!DataType Then
```

```
                    Child("DataName").Value = Rst.Fields("TypeName")

                End If
                Rst.MoveNext
            Loop
        Next Child

        dbs.Close

        Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\Net.mdb")
        dbs.Execute"Drop table ChildInfo"
        dbs.Close

    End Select
End Sub
```

```
Public Sub AA_ANN_OnSimulationInitialize(ob As CFCSim_ModelingElementInstance)
    ob.AddEvent "Observe",True
    ob.AddEvent "CheckNet"
End Sub
```

```
Public Sub AA_ANN_OnSimulationInitializeRun(ob As
CFCSim_ModelingElementInstance, RunNum As Integer)
```

```
	Dim ent As CFCSim_Entity
	Set ent = ob.AddEntity

	ob.ScheduleEvent ent,"Observe",ob("FirstObs").Value
	Tracer.Trace "First Obs for Net Fired at " & ob("FirstObs").Value, "Net"
End Sub
```

```
Public Sub AA_ANN_OnSimulationProcessEvent(ob As
CFCSim_ModelingElementInstance, MyEvent As String, Entity As CFCSim_Entity)

	Dim Child As CFCSim_ModelingElementInstance
	Dim CvrtOutput As Single
	Dim DataNum As Integer
	Dim dbs As Database
	Dim ent3 As CFCSim_Entity
	Dim ent4 As CFCSim_Entity
	Dim I As Integer
	Dim J As Integer
	Dim MaxVal As Single
	Dim MinVal As Single
	Dim Net As Integer
	Dim NormedVal As Single
	Dim Observe_Input() As Single
Dim Observe_Output() As Single
Dim OutEnt As CFCSim_Entity
Dim PlaceHolder As Integer
Dim RecCount As Integer
Dim RecLoop As Integer
Dim RecMaxNum As Integer
Dim RecMinNum As Integer
Dim Rst As Recordset
Dim Rst1 As Recordset
Dim Rst2 As Recordset
Dim SmallerX As Integer
Dim SQLString As String
Dim Table As TableDef
Dim TotalVars As Integer
Dim Training_Input() As Single
Dim Training_Output() As Single
Dim X As Integer
Dim Y As Integer
Dim GetResults As New NeuralCalc

        Dim W As Single

Select Case MyEvent
```

```
Case "Observe"


        'Drop the temp tables from the db

        Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\net.mdb")

        For Each Table In dbs.TableDefs

                If Table.Name = "Vector" Then dbs.Execute "Drop Table Vector"

                If Table.Name = "DataList" Then dbs.Execute "Drop Table DataList"

                If Table.Name = "NetOut" Then dbs.Execute "Drop Table NetOut"

                If Table.Name = "NetIn" Then dbs.Execute "Drop Table NetIn"

        Next Table

        dbs.Close


        SQLString = "Create Table Vector (RecordNo  integer,"

        For X = 1 To (ob("NumInputs") + ob("NumOutputs"))

           SQLString = SQLString & "DataType" & X & " text(55),"

           SQLString = SQLString & "DataName" & X & " text(55),"

           SQLString = SQLString & "DataValue" & X & " text(55),"

           SQLString = SQLString & "MinDataValue" & X & " text(55),"

           SQLString = SQLString & "MaxDataValue" & X & " text(55),"

        Next X


        SQLString = SQLString & "LastCol integer)"
        Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\net.mdb")
        dbs.Execute SQLString
        dbs.Close

        ' add the data to the table
        ' first get the individual data values
        Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\net.mdb")
        SQLString = "SELECT NetData.RecordNo, NetDataInfo.TypeName,
NetDataInfo.DataType, NetData.DataValue INTO DataList FROM Net INNER JOIN
(NetDataInfo INNER JOIN NetData ON NetDataInfo.NetDataInfoID =
NetData.NetDataInfoID) ON Net.NetID = NetDataInfo.NetID Where(net.NetID) = " &
ob("NetDBID")
        dbs.Execute SQLString
        dbs.Close

        'find the min and max record numbers
        Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\net.mdb")
        Set Rst1 = dbs.TableDefs("DataList").OpenRecordset

        If Rst1.RecordCount > 0 Then
            Rst1.MoveFirst
            RecMinNum = Rst1!recordNo
            RecMaxNum = Rst1!recordNo
```

```
    Do Until Rst1.EOF
      If Rst1!recordNo < RecMinNum Then RecMinNum = Rst1!recordNo
      If Rst1!recordNo > RecMaxNum Then RecMaxNum = Rst1!recordNo
      Rst1.MoveNext
    Loop
  End If

  'now sort out the values into records
  Set Rst2 = dbs.TableDefs("Vector").OpenRecordset

  For RecLoop = RecMinNum To RecMaxNum
    Rst2.AddNew
    Rst2!recordNo = RecLoop
    Rst1.MoveFirst
    Do Until Rst1.EOF
      If Rst1!recordNo = RecLoop Then
        For X = 1 To Val(ob("NumInputs"))
          If Rst1!DataType = "Input" & X Then
            Rst2.Fields("DataType" & X) = Rst1!DataType
            Rst2.Fields("DataName" & X) = Rst1.Fields("TypeName")
            Rst2.Fields("DataValue" & X) = Rst1!DataValue
          End If
        Next X
      Y = X
      For X = 1 To Val(ob("NumOutputs"))
        If Rst1!DataType = "Output" & X Then
          PlaceHolder = X + Val(ob("NumInputs"))
          Rst2.Fields("DataType" & PlaceHolder) = Rst1!DataType
          Rst2.Fields("DataName" & PlaceHolder) =
Rst1.Fields("TypeName")
          Rst2.Fields("DataValue" & PlaceHolder) = Rst1!DataValue
        End If
      Next X
    End If
    Rst1.MoveNext
  Loop
  Rst2.Update
Next RecLoop
dbs.Close

'Normalize the data (net needs the stuff to be between 0 and 1)

Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\net.mdb")
Set Rst = dbs.TableDefs("Vector").OpenRecordset

DataNum = Val(ob("NumInputs")) + Val(ob("NumOutputs"))
```

```
            For X = 1 To DataNum
              Rst.MoveFirst
              MinVal = Rst.Fields("DataValue" & X)
              MaxVal = Rst.Fields("DataValue" & X)
              Do Until Rst.EOF
                If Rst.Fields("DataValue" & X) > MaxVal Then MaxVal =
Rst.Fields("DataValue" & X)
                If Rst.Fields("DataValue" & X) < MinVal Then MinVal =
Rst.Fields("DataValue" & X)
                Rst.MoveNext
              Loop
                  For Each Child In ob.ChildElements
                      If X <= Val(ob("NumInputs")) Then
                          If Child("DataType") = "Input" & X Then
                              If IsNumeric(Child("ImposedMax")) = True
Then
                                  If Child("ImposedMax") > MaxVal
Then MaxVal =Child("ImposedMax")
                              End If
                              If IsNumeric(Child("ImposedMin")) = True
Then
                                  If Child("ImposedMin") < MinVal Then
MinVal =Child("ImposedMin")
```

```
                        End If
                    End If
                ElseIf X > Val(ob("NumInputs")) Then
                    SmallerX = X-Val(ob("NumInputs"))
                    If Child("DataType") = "Output" & SmallerX Then
                        If   IsNumeric(Child("ImposedMax"))   =   True
Then
                            If   Child("ImposedMax")   >   MaxVal
Then MaxVal =Child("ImposedMax")
                        End If
                        If   IsNumeric(Child("ImposedMin"))   =   True
Then
                            If Child("ImposedMin") < MinVal Then
MinVal =Child("ImposedMin")
                        End If
                    End If
                End If

            Next Child
            Rst.MoveFirst
        Do Until Rst.EOF
          ' norm = (x-min)/(max-min)
          ' x = norm*(max-min)+min
```

```
                Rst.Edit
                Rst.Fields("DataValue" & X) = (Rst.Fields("DataValue" & X) - MinVal)
/ (MaxVal - MinVal)
                Rst.Fields("MinDataValue" & X) = MinVal
                Rst.Fields("MaxDataValue" & X) = MaxVal
                Rst.Update
                Rst.MoveNext
              Loop
            Next
            dbs.Close

            'pass the min/max values onto the child elements so the inputs and outputs can
be scaled
            Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\net.mdb")
            Set Rst = dbs.TableDefs("Vector").OpenRecordset
            Rst.MoveFirst

            For X = 1 To DataNum
                  For Each Child In ob.ChildElements
                        If Child("DataType") = Rst.Fields("DataType" & X) Then
                              Child("MinVal")   =   Rst.Fields("MinDataValue"   &
X).Value
```

```
                        Child("MaxVal")  =  Rst.Fields("MaxDataValue"  &
X).Value
                  End If
              Next Child
          Next X
          dbs.Close


          'Schedule the next observation and training of the net
          Set ent3 = ob.AddEntity
          Set ent4 = ob.AddEntity

          ob.ScheduleEvent ent3,"Observe", ob("ObservationInterval")
          ob.ScheduleEvent ent4,"CheckNet", 0


      Case"CheckNet"

          ReDim Observe_Input(1 To Val(ob("NumInputs")), 1 To 1)
          ReDim Observe_Output(1 To Val(ob("NumOutputs")), 1 To 1)

          'get the input array
          For X = 1 To ob("NumInputs")
```

```
                    For Each Child In ob.ChildElements
                        If Child("DataType") = "Input" & X Then
                            NormedVal                    =(Child("Obsvalue")-
Child("MinVal"))/(Child("MaxVal")-Child("MinVal"))
                            Observe_Input(X,1) = NormedVal
                        End If
                    Next Child
                Next X

    Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\net.mdb")
    SQLString = "Create Table NetIn(InVal double)"
            dbs.Execute SQLString
    dbs.Close

            Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\net.mdb")
    Set Rst = dbs.TableDefs("NetIn").OpenRecordset
    For X = 1 To ob("NumInputs")
      Rst.AddNew
        Rst!InVal = Observe_Input(X, 1)
      Rst.Update
    Next X
            dbs.Close
            W                                                              =
GetResults.NeuralCalc(Val(ob("NumInputs")),Val(ob("NumOutputs")),Val(ob("AverageErr
or")), Val(ob("LearnRate")),Val(ob("Momentum")))

            'put the info into the child outputs
            Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\Net.mdb")
            Set Rst = dbs.TableDefs("NetOut").OpenRecordset
            Rst.MoveFirst

            For X = 1 To Val(ob("NumOutputs"))
                For Each Child In ob.ChildElements
                    If Child("DataType") = "Output" & X Then
                        CvrtOutput    =    Rst!OutVal*    (Child("MaxVal")-
Child("MinVal"))+ Val(Child("MinVal"))
                Child("ObsValue") = CvrtOutput
                Set OutEnt = ob.AddEntity
                        Child.ScheduleEvent OutEnt,"ProcessEffect", 0.0001
              End If
            Next Child
                If Rst.EOF= False Then Rst.MoveNext
            Next X
            dbs.Close
```

```
        Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\net.mdb")
        dbs.Execute "Drop Table NetOut"
        dbs.Close

    End Select

End Sub
```

A9.2.2 Condition Element Programming Code

```
Option Explicit
```

```
Public Function Condition_ANN_OnCreate(ob As CFCSim_ModelingElementInstance, x As Single, y As Single) As Boolean
    Condition_ANN_OnCreate=True
    ob.OnCreate x,y,True

    ob.AddAttribute "DataType","Data Type",CFC_Text,CFC_Single,CFC_ReadOnly
    ob.AddAttribute "DataName","Data Name",CFC_Text,CFC_Single,CFC_ReadOnly
    ob.AddAttribute "ObsValue","Input Variable To Observe (Linked Value)",CFC_Text,CFC_Single,CFC_ReadWrite
```

```
	ob.AddAttribute "ImposedMax","Imposed Maximum For Data Normalization - 'N/A' for not applicable (Optional - Can be done based on training data)",CFC_Text,CFC_Single,CFC_ReadWrite
	ob.AddAttribute "ImposedMin","Imposed Minimum For Data Normalization - 'N/A' for not applicable (Optional - Can be done based on training data)",CFC_Text,CFC_Single,CFC_ReadWrite
	ob.AddAttribute "MinVal","Calculated Normalization Minimum From Database",CFC_Numeric,CFC_Single,CFC_Hidden
	ob.AddAttribute "MaxVal","Calculated Normalization Maximum From Database",CFC_Numeric,CFC_Single,CFC_Hidden

	ob("ImposedMax") = "N/A"
	ob("ImposedMin") = "N/A"

	ob.SetNumCoordinates 2
	ob.CoordinatesX(0)=x
	ob.CoordinatesY(0)=y
	ob.CoordinatesX(1)=x+80
	ob.CoordinatesY(1)=y+80
End Function
```

```
Public Sub Condition_ANN_OnDragDraw(ob As CFCSim_ModelingElementInstance)
	ob.OnDraw
```

```
End Sub
```

```
Public Sub Condition_ANN_OnDraw(ob As CFCSim_ModelingElementInstance)
        CDC.RenderPicture
"if.bmp",ob.CoordinatesX(0),ob.CoordinatesY(0),ob.CoordinatesX(1)-
ob.CoordinatesX(0),ob.CoordinatesY(1)-ob.CoordinatesY(0)

        If ob.Selected Then
                CDC.ChangeLineStyle CFC_DOT,1,RGB(255,0,0)
                CDC.Rectangle ob.CoordinatesX(0)-2,ob.CoordinatesY(0)-
2,ob.CoordinatesX(1)+2,ob.CoordinatesY(1)+2
        End If

        CDC.ChangeLineStyle CFC_SOLID,1,RGB(0,0,0)
End Sub
```

```
Public Sub Condition_ANN_OnSimulationInitialize(ob As
CFCSim_ModelingElementInstance)
        ob.AddEvent "CheckModel",True
End Sub
```

```
Public Sub Condition_ANN_OnSimulationProcessEvent(ob As
CFCSim_ModelingElementInstance, MyEvent As String, Entity As CFCSim_Entity)
```

```
Dim x As String
        Select Case MyEvent
                Case "CheckRule"
                                x = ob("ObsValue")
        End Select
End Sub
```

A9.2.3 Effect Element Programming Code

```
Option Explicit
```

```
Public Function Effect_ANN_OnCreate(ob As CFCSim_ModelingElementInstance, X As
Single, y As Single) As Boolean
        Effect_ANN_OnCreate=True
        ob.OnCreate X,y,True

        ob.AddAttribute "DataType","Data Type",CFC_Text,CFC_Single,CFC_ReadOnly
        ob.AddAttribute "DataName","Data Name",CFC_Text,CFC_Single,CFC_ReadOnly
        ob.AddAttribute "ObsValue","Current Output
Value",CFC_Text,CFC_Single,CFC_ReadWrite
        ob.AddAttribute "Evaluation","Evaluation Value - if reference is needed to model
component",CFC_Text,CFC_Single,CFC_ReadWrite
```

```
    ob.AddAttribute "Effect","Effect (Linked Value)",CFC_Text,CFC_Single,CFC_ReadWrite
    ob.AddAttribute "ImposedMax","Imposed Maximum For Data Normalization - 'N/A' for not applicable (Optional - Can be done based on training data)",CFC_Text,CFC_Single,CFC_ReadWrite
    ob.AddAttribute "ImposedMin","Imposed Minimum For Data Normalization - 'N/A' for not applicable (Optional - Can be done based on training data)",CFC_Text,CFC_Single,CFC_ReadWrite

    ob.AddAttribute "MinVal","Calculated Normalization Minimum From Database",CFC_Numeric,CFC_Single,CFC_Hidden
    ob.AddAttribute "MaxVal","Calculated Normalization Maximum From Database",CFC_Numeric,CFC_Single,CFC_Hidden

    ob("ImposedMax") = "N/A"
    ob("ImposedMin") = "N/A"

    ob.SetNumCoordinates 2
    ob.CoordinatesX(0)=X
    ob.CoordinatesY(0)=y
    ob.CoordinatesX(1)=X+80
    ob.CoordinatesY(1)=y+80
End Function
```

```
Public Sub Effect_ANN_OnDragDraw(ob As CFCSim_ModelingElementInstance)
        ob.OnDraw
End Sub
```

```
Public Sub Effect_ANN_OnDraw(ob As CFCSim_ModelingElementInstance)
        CDC.RenderPicture
"then.bmp",ob.CoordinatesX(0),ob.CoordinatesY(0),ob.CoordinatesX(1)-
ob.CoordinatesX(0),ob.CoordinatesY(1)-ob.CoordinatesY(0)

        If ob.Selected Then
                CDC.ChangeLineStyle CFC_DOT,1,RGB(255,0,0)
                CDC.Rectangle                   ob.CoordinatesX(0)-2,ob.CoordinatesY(0)-
2,ob.CoordinatesX(1)+2,ob.CoordinatesY(1)+2
        End If

        CDC.ChangeLineStyle CFC_SOLID,1,RGB(0,0,0)
End Sub
```

```
Public Sub Effect_ANN_OnSimulationInitialize(ob As CFCSim_ModelingElementInstance)
        ob.AddEvent "ProcessEffect",True
End Sub
```

```
Public Sub Effect_ANN_OnSimulationProcessEvent(ob As CFCSim_ModelingElementInstance, MyEvent As String, Entity As CFCSim_Entity)
    Dim X As String

    Select Case MyEvent
    Case "ProcessEffect"
            X = ob("Effect")
    End Select
End Sub
```

A9.2.4 Dynamic Library Link

```
Option Explicit
```

```
Public Function GetNextNet(net As Integer) As Integer
  GetNextNet = NW32.GetNextNet(net)
End Function
```

```
Public Function MakeNet(net As Integer, NetType As Integer, s As Long) As Integer
  MakeNet = NW32.MakeNet(net, NetType, s)
End Function
```

```
Public Function MakeSlab(net As Integer, slab As Integer, Neurons As Integer, layer As Integer) As Integer
    MakeSlab = NW32.MakeSlab(net, slab, Neurons, layer)
End Function
```

```
Public Function MakeLink(net As Integer, Link As Integer, Weight As Single, fromslab As Integer, toslab As Integer) As Integer
    MakeLink = NW32.MakeLink(net, Link, Weight, fromslab, toslab)
End Function
```

```
Public Function PutSlab(net As Integer, slab As Integer, vector As Single) As Integer
    PutSlab = NW32.PutSlab(net, slab, vector)
End Function
```

```
Public Function GetSlab(net As Integer, slab As Integer, vector As Single) As Integer
    GetSlab = NW32.GetSlab(net, slab, vector)
End Function
```

```
Public Function PutSlabInteger(net As Integer, slab As Integer, vector As Integer) As Integer
    PutSlabInteger = NW32.PutSlabInteger(net, slab, vector)
End Function
```

```
Public Function GetSlabInteger(net As Integer, slab As Integer, vector As Integer) As Integer
```

```
   GetSlabInteger = NW32.GetSlabInteger(net, slab, vector)
End Function
```

```
Public Function PutNeuron(net As Integer, slab As Integer, neuron As Integer, Value As Single) As Integer
   PutNeuron = NW32.PutNeuron(net, slab, neuron, Value)
End Function
```

```
Public Function GetNeuron(net As Integer, slab As Integer, neuron As Integer, Value As Single) As Integer
   GetNeuron = NW32.GetNeuron(net, slab, neuron, Value)
End Function
```

```
Public Function PutLink(net As Integer, Link As Integer, vector As Single) As Integer
   PutLink = NW32.PutLink(net, Link, vector)
End Function
```

```
Public Function GetLink(net As Integer, Link As Integer, vector As Single) As Integer
   GetLink = NW32.GetLink(net, Link, vector)
End Function
```

```
Public Function PutWeight(net As Integer, Link As Integer, FromNeuron As Integer, ToNeuron As Integer, Weight As Single) As Integer
```

```
    PutWeight = NW32.PutWeight(net, Link, FromNeuron, ToNeuron, Weight)
End Function
```

```
Public Function GetWeight(net As Integer, Link As Integer, FromNeuron As Integer,
ToNeuron As Integer, Weight As Single) As Integer
    GetWeight = NW32.GetWeight(net, Link, FromNeuron, ToNeuron, Weight)
End Function
```

```
Public Function PutLinkChanges(net As Integer, Link As Integer, vector As Single) As
Integer
    PutLinkChanges = NW32.PutLinkChanges(net, Link, vector)
End Function
```

```
Public Function GetLinkChanges(net As Integer, Link As Integer, vector As Single) As
Integer
    GetLinkChanges = NW32.GetLinkChanges(net, Link, vector)
End Function
```

```
Public Function PutChange(net As Integer, Link As Integer, FromNeuron As Integer,
ToNeuron As Integer, change As Single) As Integer
    PutChange = NW32.PutChange(net, Link, FromNeuron, ToNeuron, change)
End Function
```

```
Public Function GetChange(net As Integer, Link As Integer, FromNeuron As Integer, ToNeuron As Integer, change As Single) As Integer
   GetChange = NW32.GetChange(net, Link, FromNeuron, ToNeuron, change)
End Function
```

```
Public Function BpPropagate(net As Integer, toslab As Integer) As Integer
   BpPropagate = NW32.BpPropagate(net, toslab)
End Function
```

```
Public Function BpEvaluate(net As Integer, slab As Integer, actuals As Single, errorfactor As Single) As Integer
   BpEvaluate = NW32.BpEvaluate(net, slab, actuals, errorfactor)
End Function
```

```
Public Function BpEvaluateSum(net As Integer, slab As Integer, actuals As Single) As Integer
   BpEvaluateSum = NW32.BpEvaluateSum(net, slab, actuals)
End Function
```

```
Public Function BpGetSum(net As Integer, slab As Integer, errorfactor As Single) As Integer
   BpGetSum = NW32.BpGetSum(net, slab, errorfactor)
End Function
```

```
Public Function BpEvaluateInteger(net As Integer, slab As Integer, actuals As Integer, errorfactor As Single) As Integer
   BpEvaluateInteger = NW32.BpEvaluateInteger(net, slab, actuals, errorfactor)
End Function
```

```
Public Function BpTrain(net As Integer, fromslab As Integer, learnrate As Single, Momentum As Single, updateswitch As Integer) As Integer
   BpTrain = NW32.BpTrain(net, fromslab, learnrate, Momentum, updateswitch)
End Function
```

```
Public Function BpEvaluateStore(net As Integer, slab As Integer, net2 As Integer, Store As Integer, Index As Long, errorfactor As Single) As Integer
   BpEvaluateStore = NW32.BpEvaluateStore(net, slab, net2, Store, Index, errorfactor)
End Function
```

```
Public Function BpEvaluateSumStore(net As Integer, slab As Integer, net2 As Integer, Store As Integer, Index As Long) As Integer
   BpEvaluateSumStore = NW32.BpEvaluateSumStore(net, slab, net2, Store, Index)
End Function
```

```
Public Function LvqPropagate(net As Integer, toslab As Integer) As Integer
   LvqPropagate = NW32.LvqPropagate(net, toslab)
End Function
```

```
Public Function LvqEvaluate(net As Integer, slab As Integer, Winner As Integer, adjust As
Integer) As Integer
  LvqEvaluate = NW32.LvqEvaluate(net, slab, Winner, adjust)
End Function
```

```
Public Function LvqTrain(net As Integer, fromslab As Integer, learnrate As Single,
neighborhood As Integer, Cols As Integer) As Integer
  LvqTrain = NW32.LvqTrain(net, fromslab, learnrate, neighborhood, Cols)
End Function
```

```
Public Function PnnPropagate(net As Integer, toslab As Integer, Smoothing As Single) As
Integer
  PnnPropagate = NW32.PnnPropagate(net, toslab, Smoothing)
End Function
```

```
Public Function PnnEvaluate(net As Integer, slab As Integer, actuals As Single, pattern As
Integer) As Integer
  PnnEvaluate = NW32.PnnEvaluate(net, slab, actuals, pattern)
End Function
```

```
Public Function PnntTrainingRecord(net As Integer, fromslab As Integer, pattern As Integer)
As Integer
```

```
  PnntTrainingRecord = NW32.PnntTrainingRecord(net, fromslab, pattern)
End Function
```

```
Public Function PutCounts(net As Integer, slab As Integer, vector As Single) As Integer
  PutCounts = NW32.PutCounts(net, slab, vector)
End Function
```

```
Public Function GetCounts(net As Integer, slab As Integer, vector As Single) As Integer
  GetCounts = NW32.GetCounts(net, slab, vector)
End Function
```

```
Public Function GrnnPropagate(net As Integer, toslab As Integer, Smoothing As Single) As Integer
  GrnnPropagate = NW32.GrnnPropagate(net, toslab, Smoothing)
End Function
```

```
Public Function GrnnEvaluate(net As Integer, slab As Integer, actuals As Single, pattern As Integer) As Integer
  GrnnEvaluate = NW32.GrnnEvaluate(net, slab, actuals, pattern)
End Function
```

```
Public Function GrnntTrainingRecord(net As Integer, fromslab As Integer, pattern As Integer) As Integer
```

```
  GrnntTrainingRecord = NW32.GrnntTrainingRecord(net, fromslab, pattern)
End Function
```

```
Public Function CopySlab(fromnet As Integer, fromslab As Integer, tonet As Integer, toslab As Integer) As Integer
  CopySlab = NW32.CopySlab(fromnet, fromslab, tonet, toslab)
End Function
```

```
Public Function SetSlab(net As Integer, slab As Integer, Value As Single) As Integer
  SetSlab = NW32.SetSlab(net, slab, Value)
End Function
```

```
Public Function SlideSlab(net As Integer, slab As Integer, Value As Single) As Integer
  SlideSlab = NW32.SlideSlab(net, slab, Value)
End Function
```

```
Public Function InitLink(net As Integer, Link As Integer, Weight As Single) As Integer
  InitLink = NW32.InitLink(net, Link, Weight)
End Function
```

```
Public Function KillNet(net As Integer) As Integer
  KillNet = NW32.KillNet(net)
End Function
```

```
Public Function KillSlab(net As Integer, slab As Integer) As Integer
  KillSlab = NW32.KillSlab(net, slab)
End Function
```

```
Public Function KillLink(net As Integer, Link As Integer) As Integer
  KillLink = NW32.KillLink(net, Link)
End Function
```

```
Public Function KillStore(net As Integer, Store As Integer) As Integer
  KillStore = NW32.KillStore(net, Store)
End Function
```

```
Public Function SetMath(mathtype As Integer) As Integer
  SetMath = NW32.SetMath(mathtype)
End Function
```

```
Public Function SetCPU(cpu As Integer) As Integer
  SetCPU = NW32.SetCPU(cpu)
End Function
```

```
Public Function LoadShell(net As Integer, ByVal path As String) As Integer
  LoadShell = NW32.LoadShell(net, ByVal path)
```

```
End Function
```

```
Public Function MakeStore(net As Integer, Store As Integer, slabsize As Integer, numslabs
As Long) As Integer
  MakeStore = NW32.MakeStore(net, Store, slabsize, numslabs)
End Function
```

```
Public Function PutStore(net As Integer, Store As Integer, Index As Long, vector As Single)
As Integer
  PutStore = NW32.PutStore(net, Store, Index, vector)
End Function
```

```
Public Function GetStore(net As Integer, Store As Integer, Index As Long, vector As Single)
As Integer
  GetStore = NW32.GetStore(net, Store, Index, vector)
End Function
```

```
Public Function CopyToStore(net As Integer, Store As Integer, Index As Long, net2 As
Integer, slab As Integer) As Integer
  CopyToStore = NW32.CopyToStore(net, Store, Index, net2, slab)
End Function
```

```
Public Function CopyFromStore(net As Integer, Store As Integer, Index As Long, net2 As Integer, slab As Integer) As Integer
  CopyFromStore = NW32.CopyFromStore(net, Store, Index, net2, slab)
End Function
```

```
Public Function AddSlab(net As Integer, fromslab As Integer, toslab As Integer, factor As Single) As Integer
  AddSlab = NW32.AddSlab(net, fromslab, toslab, factor)
End Function
```

```
Public Function SaveNet(net As Integer, ByVal path As String, usersareasize As Long, userarea As Single) As Integer
  SaveNet = NW32.SaveNet(net, ByVal path, usersareasize, userarea)
End Function
```

```
Public Function RestoreNet(net As Integer, ByVal path As String, s As Long, usersareasize As Long, userarea As Single) As Integer       ' s is your NeuroWin serial number
  RestoreNet = NW32.RestoreNet(net, ByVal path, s, usersareasize, userarea)
End Function
```

```
Public Function Transfer(net As Integer, slab As Integer, activation As Integer) As Integer
  Transfer = NW32.Transfer(net, slab, activation)
End Function
```

```
Public Function SetSeed(seed As Integer) As Integer
  SetSeed = NW32.SetSeed(seed)
End Function
```

```
Public Function Map(ByVal Value As Single, min As Single, max As Single, x1 As Single, x2 As Single) As Single
   Map = NW32.Map(ByVal Value, min, max, x1, x2)
End Function
```

```
Public Function NetError(ErrorNum As Integer) As String
   NetError = NW32.NetError(ErrorNum)
End Function
```

```
Public Sub Normalize(vector() As Single, Count As Integer)
   NW32.Normalize vector(), Count
End Sub
```

```
Public Function SD3(vector() As Single, Count As Integer, mean As Single) As Single
   SD3 = NW32.SD3(vector(), Count, mean)
End Function
```

```
Public Function Winner(vector() As Single, Count As Integer) As Integer
```

```
 Winner = NW32.Winner(vector(), Count)
End Function
```

A9.2.5 Observation Assistant Element Programming Code

```
Option Explicit
```

```
Public Function AA_ANN_O_OnCreate(ob As CFCSim_ModelingElementInstance, X As
Single, Y As Single) As Boolean
        AA_ANN_O_OnCreate=True
        ob.OnCreate X,Y,True

        ob.AddAttribute "NeuralFile","Neural Network
File",CFC_Text,CFC_ListBox,CFC_ReadWrite
        ob.AddAttribute "FirstObs","Time of First
Observation",CFC_Numeric,CFC_Single,CFC_ReadWrite
        ob.AddAttribute "ObservationInterval","Observation
Interval",CFC_Numeric,CFC_Single,CFC_ReadWrite

        ob.AddAttribute "NetDBID","Network DB ID",CFC_Text,CFC_Single,CFC_Hidden
        ob.AddAttribute "NumInputs","Number of Input
Nodes",CFC_Numeric,CFC_Single,CFC_Hidden
```

```
    ob.AddAttribute "NumOutputs","Number of Output Nodes",CFC_Numeric,CFC_Single,CFC_Hidden
    ob.AddAttribute "NextRecNo","Next Record Number",CFC_Numeric,CFC_Single,CFC_Hidden

    ob.Attr("ObservationInterval") = 480
    ob.Attr("FirstObs") = 480

    ob.SetNumCoordinates 2
    ob.CoordinatesX(0)=X
    ob.CoordinatesY(0)=Y
    ob.CoordinatesX(1)=X+70
    ob.CoordinatesY(1)=Y+49

End Function
```

```
Public Sub AA_ANN_O_OnDragDraw(ob As CFCSim_ModelingElementInstance)
    ob.OnDraw
End Sub
```

```
Public Sub AA_ANN_O_OnDraw(ob As CFCSim_ModelingElementInstance)
```

```
        CDC.RenderPicture
"harrys1.bmp",ob.CoordinatesX(0),ob.CoordinatesY(0),ob.CoordinatesX(1)-
ob.CoordinatesX(0),ob.CoordinatesY(1)-ob.CoordinatesY(0)

        If ob.Selected Then
                CDC.ChangeLineStyle CFC_DOT,1,RGB(255,0,0)
                CDC.Rectangle ob.CoordinatesX(0)-2,ob.CoordinatesY(0)-
2,ob.CoordinatesX(1)+2,ob.CoordinatesY(1)+2
        End If

        CDC.ChangeLineStyle CFC_SOLID,1,RGB(0,0,0)

End Sub
```

```
Public Sub AA_ANN_O_OnListBoxInitialize(ob As CFCSim_ModelingElementInstance,
attr As CFCSim_Attribute, lstList As Object)
        Dim dbs As Database
        Dim Rst As Recordset

        Select Case attr.Name
        Case "NeuralFile"
                Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\Net.mdb")
                Set Rst = dbs.TableDefs("Net").OpenRecordset
```

```
                    If Rst.RecordCount > 0 Then
                    Rst.MoveFirst
                    Do Until Rst.EOF
                    lstList.additem Rst!NetName
                    Rst.MoveNext
                    Loop
                    End If
          End Select
End Sub
```

```
Public Sub AA_ANN_O_OnListBoxSelectItem(ob As CFCSim_ModelingElementInstance,
attr As CFCSim_Attribute, lstList As Object)
          Dim dbs As Database
          Dim Rst As Recordset
          Dim Child As CFCSim_ModelingElementInstance
          Dim X As Integer
          Dim Table As TableDef

          attr.Value = lstList.text
          ob(attr.Name) = attr.Value
          Select Case attr.Name
          Case "NeuralFile"
```

```
'clean out db if old table exists
Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\net.mdb")
For Each Table In dbs.TableDefs
        If Table.Name = "ChildInfo" Then dbs.Execute "Drop Table ChildInfo"
Next Table
dbs.Close

Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\Net.mdb")
Set Rst = dbs.TableDefs("Net").OpenRecordset
'get the net info
If Rst.RecordCount > 0 Then
        Rst.MoveFirst
        Do Until Rst.EOF
                If ob("NeuralFile").Value = Rst!NetName Then
                        ob.Attr("NetDBID").Value = Rst!NetId
                        ob.Attr("NumInputs").Value = Rst!InputNum
                        ob.Attr("NumOutputs").Value = Rst!OutputNum
                End If
                Rst.MoveNext
Loop
End If
'clean out the child elements if a new net is selected
```

```
If ob.ChildElements.Count> 0 Then
        For Each Child In ob.ChildElements
                Child.Delete
        Next Child
End If
'put in the elements
Dim NewEle As CFCSim_ModelingElementInstance

For X = 1 To ob("NumInputs")
        Set NewEle = ob.AddElement("Observer_ANN",X*100, 50)
        NewEle("DataType") = "Input" & X
Next X

For X = 1 To ob("NumOutputs")
        Set NewEle = ob.AddElement("Observer_ANN",X*100, 250)
        NewEle("DataType") = "Output" & X
Next X

'feed the info into the new elements
dbs.Execute"Select NetDataInfo.DataType, NetDataInfo.TypeName INTO
ChildInfo FROM Net INNER Join NetDataInfo On Net.NetID = NetDataInfo.NetID
WHERE (Net.NetID)=" & ob("NetDBID")
dbs.Close
```

```
        Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\Net.mdb")
        Set Rst = dbs.TableDefs("ChildInfo").OpenRecordset

        For Each Child In ob.ChildElements
            Rst.MoveFirst
            Do Until Rst.EOF
                If Child("DataType") = Rst!DataType Then
                    Child("DataName").Value = Rst!TypeName
                End If
                Rst.MoveNext
            Loop
        Next Child

        dbs.Close

        Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\Net.mdb")
        dbs.Execute"Drop table ChildInfo"
        dbs.Close

    End Select
End Sub
```

```
Public Sub AA_ANN_O_OnSimulationInitialize(ob As CFCSim_ModelingElementInstance)

        ob.AddEvent "Observe",True

        ob.AddEvent "RecordData"

End Sub
```

```
Public Sub AA_ANN_O_OnSimulationInitializeRun(ob As
CFCSim_ModelingElementInstance, RunNum As Integer)

        Dim ent As CFCSim_Entity

        Set ent = ob.AddEntity


        ob.ScheduleEvent ent,"Observe",ob("FirstObs")


End Sub
```

```
Public Sub AA_ANN_O_OnSimulationProcessEvent(ob As
CFCSim_ModelingElementInstance, MyEvent As String, Entity As CFCSim_Entity)

        Select Case MyEvent

        Case "Observe"


                'get each child to observe the model

                Dim Child As CFCSim_ModelingElementInstance


                For Each Child In ob.ChildElements
```

```
            Dim ent As CFCSim_Entity

            Set ent = ob.AddEntity

            Child.ScheduleEvent ent,"FindDataID",0

            Tracer.Trace "Event Schedulde for " & Child("DataType"), "Sim",

"Scheduled"

        Next Child


        'find the next record (or Vector) number in the db

        Dim dbs As Database

        Dim Rst As Recordset

        Dim HighRecNo As Integer


        Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\Net.mdb")

        Set Rst = dbs.TableDefs("NetData").OpenRecordset

        Rst.MoveFirst

        HighRecNo = Rst!RecordNo


        Do Until Rst.EOF

            If HighRecNo < Rst!RecordNo Then HighRecNo = Rst!RecordNo

            Rst.MoveNext

        Loop

        dbs.Close
        ob("NextRecNo")= HighRecNo +1


        'schedule the recording of the data

        Dim ent2 As CFCSim_Entity

        Set ent2 = ob.AddEntity

        ob.ScheduleEvent ent2,"RecordData",1


        Dim ent3 As CFCSim_Entity

        Set ent3 = ob.AddEntity

        ob.ScheduleEvent ent3,"Observe",ob("ObservationInterval")


    Case "RecordData"

        Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\Net.mdb")

        Set Rst = dbs.TableDefs("NetData").OpenRecordset


        For Each Child In ob.ChildElements

            Rst.AddNew

            Rst!NetDataInfoID = Child("DataID")

            Rst!RecordNo = ob("NextRecNo")

            Rst!DataValue = Child("ObsValue")

            Rst.Update

        Next Child

    End Select
```

```
End Sub
```

A9.2.6 Observation Element Programming Code

```
Option Explicit
```

```
Public Function Observer_ANN_OnCreate(ob As CFCSim_ModelingElementInstance, x As Single, y As Single) As Boolean
        Observer_ANN_OnCreate=True
        ob.OnCreate x,y,True

        ob.AddAttribute "DataType","Data Type",CFC_Text,CFC_Single,CFC_ReadOnly
        ob.AddAttribute "DataName","Data Name",CFC_Text,CFC_Single,CFC_ReadOnly
        ob.AddAttribute "ObsValue","Input Variable To Observe (Linked Value)",CFC_Text,CFC_Single,CFC_ReadWrite

        ob.AddAttribute "DataID", "DB Data Record ID",CFC_Numeric,CFC_Single,CFC_Hidden

        ob.SetNumCoordinates 2
        ob.CoordinatesX(0)=x
        ob.CoordinatesY(0)=y
        ob.CoordinatesX(1)=x+100
        ob.CoordinatesY(1)=y+80
End Function
```

```
Public Sub Observer_ANN_OnDragDraw(ob As CFCSim_ModelingElementInstance)
        ob.OnDraw
End Sub
```

```
Public Sub Observer_ANN_OnDraw(ob As CFCSim_ModelingElementInstance)
        CDC.RenderPicture
"Observer.bmp",ob.CoordinatesX(0),ob.CoordinatesY(0),ob.CoordinatesX(1)-
ob.CoordinatesX(0),ob.CoordinatesY(1)-ob.CoordinatesY(0)

        If ob.Selected Then
                CDC.ChangeLineStyle CFC_DOT,1,RGB(255,0,0)
                CDC.Rectangle                   ob.CoordinatesX(0)-2,ob.CoordinatesY(0)-
2,ob.CoordinatesX(1)+2,ob.CoordinatesY(1)+2
        End If

        CDC.ChangeLineStyle CFC_SOLID,1,RGB(0,0,0)
End Sub
```

```
Public          Sub          Observer_ANN_OnSimulationInitialize(ob          As
CFCSim_ModelingElementInstance)
```

```
          ob.AddEvent "FindDataID",True

          ob.AddEvent "Observe"

End Sub
```

```
Public Sub Observer_ANN_OnSimulationProcessEvent(ob As
CFCSim_ModelingElementInstance, MyEvent As String, Entity As CFCSim_Entity)

          Select Case MyEvent

          Case "FindDataID"
                    'need to find the datatype for the child element from the DB
                    Dim dbs As Database
                    Dim Rst As Recordset
                    Dim child As CFCSim_ModelingElementInstance

                    Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\Net.mdb")
                    dbs.Execute  "Select NetDataInfo.* into DataTypeList" & ob.Id & " FROM
Net INNER Join NetDataInfo On Net.NetID = NetDataInfo.NetID WHERE Net.NetID=" &
 ob.Parent("NetDBID")
                    dbs.Close

                    Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\Net.mdb")
                    Set Rst = dbs.TableDefs("DataTypeList" & ob.Id).OpenRecordset
```

```
        Rst.MoveFirst

        Do Until Rst.EOF

            If Rst!DataType = ob("DataType") Then

                ob("DataId") = Rst!NetDataInfoID

            End If

            Rst.MoveNext

        Loop

        dbs.Close



        'get rid of the temp table

        Set dbs = DBEngine.Workspaces(0).OpenDatabase("C:\NNF\Net.mdb")

        dbs.Execute  "Drop Table DataTypeList" & ob.Id

        dbs.Close


        'schedule observation

        Dim ent As CFCSim_Entity

        Set ent = ob.AddEntity

        ob.ScheduleEvent ent,"Observe",0


    Case "CheckRule"

        'initiate the linked code

        Dim x As String
```

```
            x = ob("ObsValue")
        End Select
End Sub
```

A9.3 Algorithm Agent

The following subsections provide the information pertaining to the construction of the modeling elements that make up the algorithm agent presented in this research. The algorithm presented that is used to predict artificial boreholes was developed by Ruwanpura (2000).

A9.3.1 Parent Agent Element Programming Code

```
Option Explicit
```

```
Public Function AA_DIRT_OnCreate(ob As CFCSim_ModelingElementInstance, X As Single, Y As Single) As Boolean
        AA_DIRT_OnCreate=True
        ob.OnCreate X,Y,True

        ob.AddAttribute "BoreHoleDB","BoreHole Database File",CFC_Text,CFC_ListBox,CFC_ReadWrite
        ob.AddAttribute "FirstObs","Time of First Observation",CFC_Numeric,CFC_Single,CFC_ReadWrite
        ob.AddAttribute                                "ObservationInterval","Observation
Interval",CFC_Numeric,CFC_Single,CFC_ReadWrite

        ob("BoreHoleDB") = "Borehole.mdb"
        ob("FirstObs") = 0
        ob("ObservationInterval") = 10

        ob.SetNumCoordinates 2
        ob.CoordinatesX(0)=X
        ob.CoordinatesY(0)=Y
        ob.CoordinatesX(1)=X+70
        ob.CoordinatesY(1)=Y+49
End Function
```

```
Public Sub AA_DIRT_OnDragDraw(ob As CFCSim_ModelingElementInstance)
        ob.OnDraw
End Sub
```

```
Public Sub AA_DIRT_OnDraw(ob As CFCSim_ModelingElementInstance)
        CDC.RenderPicture
"harrys1.bmp",ob.CoordinatesX(0),ob.CoordinatesY(0),ob.CoordinatesX(1)-
ob.CoordinatesX(0),ob.CoordinatesY(1)-ob.CoordinatesY(0)
```

```
        If ob.Selected Then

                CDC.ChangeLineStyle CFC_DOT,1,RGB(255,0,0)

                CDC.Rectangle                    ob.CoordinatesX(0)-2,ob.CoordinatesY(0)-

2,ob.CoordinatesX(1)+2,ob.CoordinatesY(1)+2

        End If


        CDC.ChangeLineStyle CFC_SOLID,1,RGB(0,0,0)

End Sub
```

```
Public Sub AA_DIRT_OnListBoxInitialize(ob As CFCSim_ModelingElementInstance, attr

As CFCSim_Attribute, lstList As Object)

        Dim MyFolder As New FileSystemObject

        Dim BHFolder As Folder

        Dim MyFile As File


        Select Case attr.Name

        Case "BoreHoleDB"

                Set BHFolder = MyFolder.GetFolder("C:\Borehole\")


                For Each MyFile In BHFolder.Files

                If MyFile.Type = "Microsoft Access Application" Then

                lstList.additem MyFile.Name

                End If
```

```
            Next MyFile
        End Select
End Sub
```

```
Public Sub AA_DIRT_OnSimulationInitialize(ob As CFCSim_ModelingElementInstance)
        ob.AddEvent "Observe",True
End Sub
```

```
Public Sub AA_DIRT_OnSimulationInitializeRun(ob As
CFCSim_ModelingElementInstance, RunNum As Integer)
        Dim Ent As CFCSim_Entity
        Set Ent = ob.AddEntity

        ob.ScheduleEvent Ent,"Observe",ob("FirstObs").Value
        Tracer.Trace "First Obs for Soil Prediction Fired at " & ob("FirstObs").Value, "Soil"
End Sub
```

```
Public Sub AA_DIRT_OnSimulationProcessEvent(ob As
CFCSim_ModelingElementInstance, MyEvent As String, Entity As CFCSim_Entity)
        Dim Ent1 As CFCSim_Entity

        '********************************Globals***************************
        Dim Northing As Double
Dim Easting As Double

Dim LowNorthing As Double

Dim LowEasting As Double

Dim HighNorthing As Double

Dim HighEasting As Double

Dim TheFam As String

Dim BrokenFam() As Variant'Integer

Dim Elevations() As Double


'********************************variables**************************

Dim A As Double 'a variable in x intercept eqn

Dim B As Double 'b variable in x intercept eqn

Dim BottomPart As Double

Dim C As Double 'c variable in x intercept eqn

Dim CheckDistance1 As Double

Dim CheckDistance2 As Double

Dim CurrentFam As String

Dim Child As CFCSim_ModelingElementInstance

Dim D1 As Double

Dim D2 As Double

Dim D3 As Double

Dim DB As String

Dim dbs As Database
Dim DC As Double

Dim Dist As Double

Dim E1 As Double

Dim E2 As Double

Dim E3 As Double

Dim EC As Double

Dim Ele1 As Double

Dim Ele2 As Double

Dim Ele3 As Double

Dim EleC As Double

Dim Esting As Single

Dim Fam As String

Dim FamCount As Integer

Dim FBase As Integer

Dim FNeighbour As Integer

Dim H As Double 'x center of circle

Dim HoleNo As Integer

Dim Intercept As Double 'b variable in line eqn

Dim K As Double 'y centre of circle

Dim LastSoil As Integer

Dim M As Double 'slope variable in line eqn

Dim N1 As Double

Dim N2 As Double
Dim N3 As Double
Dim NC As Double
Dim Nthing As Single
Dim OutEnt As CFCSim_Entity
Dim PredFam As String
Dim ProfileLength As Integer
Dim Radius As Double
Dim Rst As Recordset
Dim Rst2 As Recordset
Dim Rst3 As Recordset
Dim Rst4 As Recordset
Dim Soil As Integer
Dim Table As TableDef
Dim TDChange As Integer
Dim TotalDistance As Double
Dim TopPart As Double
Dim WF As Double
Dim X As Integer
Dim Y As Integer

DB = "C:\Borehole\" & ob("BoreHoleDB")
Select Case MyEvent
```

```
    Case "Observe"

    '*****************************ClearDB*****************************
        Set dbs = DBEngine.Workspaces(0).OpenDatabase(DB)

        For Each Table In dbs.TableDefs
        If Table.Name = "Fams" Then dbs.Execute "drop table fams"
          If Table.Name = "MakeHoleFams" Then dbs.Execute "drop table
makeholefams"
          If Table.Name = "AvailableFams" Then dbs.Execute "drop table
Availablefams"
          If Table.Name = "FamsCount" Then dbs.Execute "drop table FamsCount"
          If Table.Name = "PCEVals" Then dbs.Execute "drop table PCEVals"
          If Table.Name = "WFVals" Then dbs.Execute "drop table WFVals"
          If Table.Name = "ConFam" Then dbs.Execute "drop table ConFam"
          If Table.Name = "Top_Temp" Then dbs.Execute "drop table Top_Temp"
        Next Table
        dbs.Close

'***************************FindLoc********************************
        For Each Child In ob.ChildElements
            If Child.ElementType="AA_Dirt_Condition" Then
                Northing = Child("Northing")
                Easting = Child("Easting")

                Radius = Child("Radius")

            End If

        Next Child


    '****************************CalcProbs****************************

        Set dbs = DBEngine.Workspaces(0).OpenDatabase(DB)

        dbs.Execute "SELECT Boreholes.Borehole_No,
Borehole_Details.Start_Depth, Soil.Soil_No, Boreholes.Northing, Boreholes.Easting INTO
Fams FROM (Boreholes INNER JOIN Borehole_Details ON Boreholes.BoreHole_ID =
Borehole_Details.Borehole_ID) INNER JOIN Soil ON Borehole_Details.Soil_ID =
Soil.Soil_ID ORDER BY Boreholes.Borehole_No, Borehole_Details.Start_Depth DESC"

        dbs.Execute "Create Table MakeHoleFams (HoleNo integer, Fam Text(55),
Northing Single, Easting Single, Prob Single)"

        dbs.Close


        Set dbs = DBEngine.Workspaces(0).OpenDatabase(DB)

        Set Rst = dbs.TableDefs("Fams").OpenRecordset

        Set Rst2 = dbs.TableDefs("MakeHoleFams").OpenRecordset


        Rst.MoveFirst

        HoleNo = Rst!Borehole_No

        Fam = Rst!Soil_No
```

```
LastSoil = Rst!Soil_No

Nthing = Rst!Northing

Esting = Rst!Easting

Do Until Rst.EOF

If Rst!Borehole_No = HoleNo Then

If LastSoil <> Rst!Soil_No Then

        Fam = Fam & " " & Rst!Soil_No

        LastSoil = Rst!Soil_No

        Nthing = Rst!Northing

        Esting = Rst!Easting

End If

ElseIf Rst!Borehole_No <> HoleNo Then

Rst2.AddNew

Rst2!HoleNo = HoleNo

Rst2!Fam = Fam

Rst2!Northing = Nthing

Rst2!Easting = Esting

Rst2.Update

HoleNo = Rst!Borehole_No

Fam = Rst!Soil_No

LastSoil = Rst!Soil_No

Nthing = Rst!Northing

Esting = Rst!Easting
```

```
      End If
      Rst.MoveNext
      Loop
      dbs.Close
      ' delete the boreholes outside the radius
      LowNorthing = Northing - Radius
    HighNorthing = Northing + Radius
    LowEasting = Easting - Radius
    HighEasting = Easting + Radius


      Set dbs = DBEngine.Workspaces(0).OpenDatabase(DB)
      dbs.Execute "DELETE MakeHoleFams.*, MakeHoleFams.Northing From
MakeHoleFams WHERE MakeHoleFams.Northing < " & LowNorthing
      dbs.Execute "DELETE MakeHoleFams.*, MakeHoleFams.Northing From
MakeHoleFams WHERE MakeHoleFams.Northing > " & HighNorthing
      dbs.Execute "DELETE MakeHoleFams.*, MakeHoleFams.Easting From
MakeHoleFams WHERE MakeHoleFams.Easting  < " & LowEasting
      dbs.Execute "DELETE MakeHoleFams.*, MakeHoleFams.Easting From
MakeHoleFams WHERE MakeHoleFams.Easting > " & HighEasting
      dbs.Close

      Set dbs = DBEngine.Workspaces(0).OpenDatabase(DB)
```

```
        dbs.Execute "Create Table FamsCount (Fam text(55), FamCount integer)"
        dbs.Execute "SELECT MakeHoleFams.Fam Into AvailableFams From
MakeHoleFams Group By MakeHoleFams.Fam"
        dbs.Close

        Set dbs = DBEngine.Workspaces(0).OpenDatabase(DB)
        Set Rst = dbs.TableDefs("AvailableFams").OpenRecordset
        Set Rst2 = dbs.TableDefs("MakeHoleFams").OpenRecordset
        Set Rst3 = dbs.TableDefs("FamsCount").OpenRecordset

        Rst.MoveFirst
        Do Until Rst.EOF
        CurrentFam = Rst!Fam
        FamCount = 0
        Rst2.MoveFirst
        Do Until Rst2.EOF
        If Rst2!Fam = Rst!Fam Then FamCount = FamCount + 1
        Rst2.MoveNext
        Loop
        Rst3.AddNew
    Rst3!Fam = CurrentFam
    Rst3!FamCount = FamCount
        Rst3.Update
```

```
          Rst.MoveNext
          Loop
          dbs.Close

          Set dbs = DBEngine.Workspaces(0).OpenDatabase(DB)
          dbs.Execute "Create Table PCEVals (BaseFam text(55), Neighbour text(55),
BaseCount integer, NeighbourCount integer, PCE single, Distance single)"
          dbs.Close

          Set dbs = DBEngine.Workspaces(0).OpenDatabase(DB)
          Set Rst = dbs.TableDefs("AvailableFams").OpenRecordset
          Set Rst2 = dbs.TableDefs("FamsCount").OpenRecordset
          Set Rst3 = dbs.TableDefs("MakeHoleFams").OpenRecordset
          Set Rst4 = dbs.TableDefs("PCEVals").OpenRecordset

          Rst.MoveFirst
          Do Until Rst.EOF
          Rst3.MoveFirst
          Do Until Rst3.EOF
          Dist = (((Rst3!Northing - Northing) ^ 2) + ((Rst3!Easting - Easting) ^ 2)) ^
0.5
          Rst2.MoveFirst
          Do Until Rst2.EOF
```

```
    If Rst2!Fam = Rst!Fam Then FBase = Rst2!FamCount

    If Rst2!Fam = Rst3!Fam Then FNeighbour = Rst2!FamCount

    Rst2.MoveNext

Loop

Rst4.AddNew

Rst4!BaseFam = Rst!Fam

Rst4!Neighbour = Rst3!Fam

Rst4!BaseCount = FBase

Rst4!NeighbourCount = FNeighbour

If Rst!Fam = Rst3!Fam Then

Rst4!PCE = 1

Else

Rst4!PCE = FBase / (FBase + FNeighbour)

End If

Rst4!Distance = Dist

Rst4.Update

Rst3.MoveNext

Loop

Rst.MoveNext

Loop

dbs.Close


Set dbs = DBEngine.Workspaces(0).OpenDatabase(DB)
```

```
dbs.Execute "Create Table WFVals (Fam text(55), WFVal single)"

dbs.Close


Set dbs = DBEngine.Workspaces(0).OpenDatabase(DB)

Set Rst = dbs.TableDefs("AvailableFams").OpenRecordset

Set Rst2 = dbs.TableDefs("PCEVals").OpenRecordset

Set Rst3 = dbs.TableDefs("WFVals").OpenRecordset


Rst.MoveFirst

Do Until Rst.EOF

TopPart = 0

BottomPart = 0

Rst2.MoveFirst

Do Until Rst2.EOF

If Rst2!BaseFam = Rst!Fam Then

        TopPart = TopPart + Rst2!PCE / Rst2!Distance

        BottomPart = BottomPart + 1 / Rst2!Distance

End If

        Rst2.MoveNext

Loop

Rst3.AddNew

Rst3!Fam = Rst!Fam

Rst3!wfval = TopPart / BottomPart
```

```
Rst3.Update
Rst.MoveNext
Loop
dbs.Close

Set dbs = DBEngine.Workspaces(0).OpenDatabase(DB)
Set Rst = dbs.TableDefs("WFVals").OpenRecordset

Rst.MoveFirst
WF = Rst!wfval
PredFam = Rst!Fam
Do Until Rst.EOF
If Rst!wfval > WF Then
WF = Rst!wfval
PredFam = Rst!Fam
End If
Rst.MoveNext
Loop

TheFam = PredFam
ProfileLength = ((Len(TheFam) + 1) / 2) - 1
ReDim BrokenFam(0 To ProfileLength)
For X = 0 To ProfileLength
```

```
        Y = X * 2 +1
        BrokenFam(X) = Mid$(TheFam,Y,1)
    Next X

'*****************************GetDepths*****************************
    Set dbs = DBEngine.Workspaces(0).OpenDatabase(DB)
    dbs.Execute "SELECT * into ConFam FROM Fams Order By Borehole_No,
Start_Depth desc;"
    Set Rst = dbs.TableDefs("ConFam").OpenRecordset
    Set Rst2 = dbs.TableDefs("ConFam").OpenRecordset

    TDChange = 4
    Do Until TDChange < 1
        TDChange = 0
    Rst.MoveFirst
    Rst2.MoveFirst
    Rst2.MoveNext
    Do Until Rst2.EOF
    If Rst!Borehole_No = Rst2!Borehole_No Then
        If Rst!Soil_No = Rst2!Soil_No Then
        Rst2.Delete
        TDChange = TDChange + 1
        Rst2.MoveNext
    End If
End If
Rst.MoveNext
If Rst.EOF = False Then Rst2.MoveNext
Loop
Loop

dbs.Execute "SELECT ConFam.Borehole_No, Max(ConFam.Start_Depth) as Start_Depth, ConFam.Northing, ConFam.Easting into Top_Temp  FROM ConFam group by ConFam.Borehole_No, ConFam.Northing, ConFam.Easting;"
dbs.Close

Set dbs = DBEngine.Workspaces(0).OpenDatabase(DB)
Set Rst = dbs.TableDefs("Top_Temp").OpenRecordset

D1 = 9.9999E+54
D2 = 9.9999E+54
D3 = 9.9999E+54

Rst.MoveFirst
Do Until Rst.EOF
  NC = Rst!Northing
  EC = Rst!Easting
```

```
EleC = Rst!Start_Depth
DC = (((NC - Northing) ^ 2) + ((EC - Easting) ^ 2)) ^ 0.5
If DC < D1 Then
    N3 = N2
    E3 = E2
    Ele3 = Ele2
    D3 = D2

    N2 = N1
    E2 = E1
    Ele2 = Ele1
    D2 = D1

    N1 = NC
    E1 = EC
    Ele1 = EleC
    D1 = DC

ElseIf DC < D2 Then
    N3 = N2
    E3 = E2
    Ele3 = Ele2
    D3 = D2
```

```
        N2 = NC

        E2 = EC

        Ele2 = EleC

        D2 = DC

    ElseIf DC < D3 Then

        N3 = NC

        E3 = EC

        Ele3 = EleC

        D3 = DC

    End If

    Rst.MoveNext

Loop

dbs.Close

ReDim Elevations(0 To ProfileLength)

Elevations(0) = FindElev(N1, E1, Ele1, N2, E2, Ele2, N3, E3, Ele3, Northing,
Easting)
```

```
For X = 1 To ProfileLength

    Soil = BrokenFam(X)
    Y = X - 1

        Set dbs = DBEngine.Workspaces(0).OpenDatabase(DB)

    For Each Table In dbs.TableDefs
        If Table.Name = "Top_Temp" Then dbs.Execute "drop table Top_Temp"
    Next Table

        dbs.Execute "SELECT ConFam.Borehole_No, Max(ConFam.Start_Depth)
AS Start_Depth, ConFam.Northing, ConFam.Easting, ConFam.Soil_No INTO Top_Temp
FROM ConFam GROUP BY ConFam.Borehole_No, ConFam.Northing, ConFam.Easting,
ConFam.Soil_No HAVING (((Max(ConFam.Start_Depth))<" & Elevations(Y) & ") AND
((ConFam.Soil_No)=" & Soil & "));"
        dbs.Close

        Set dbs = DBEngine.Workspaces(0).OpenDatabase(DB)
    Set Rst = dbs.TableDefs("Top_Temp").OpenRecordset

    D1 = 9.9999E+54
    D2 = 9.9999E+54
```

```
D3 = 9.9999E+54

Rst.MoveFirst
Do Until Rst.EOF
    NC = Rst!Northing
    EC = Rst!Easting
    EleC = Rst!Start_Depth
    DC = (((NC - Northing) ^ 2) + ((EC - Easting) ^ 2)) ^ 0.5
    If DC < D1 Then
        N3 = N2
        E3 = E2
        Ele3 = Ele2
        D3 = D2

        N2 = N1
        E2 = E1
        Ele2 = Ele1
        D2 = D1

        N1 = NC
        E1 = EC
        Ele1 = EleC
        D1 = DC
```

```
  ElseIf DC < D2 Then
    N3 = N2
    E3 = E2
    Ele3 = Ele2
    D3 = D2

    N2 = NC
    E2 = EC
    Ele2 = EleC
    D2 = DC

  ElseIf DC < D3 Then
    N3 = NC
    E3 = EC
    Ele3 = EleC
    D3 = DC

  End If
  Rst.MoveNext
Loop
dbs.Close
```

```
                Elevations(X) = FindElev(N1, E1, Ele1, N2, E2, Ele2, N3, E3, Ele3,
Northing, Easting)
            Next X

            '**********************return profile*****************
            For Each Child In ob.ChildElements
                If Child.ElementType="AA_Dirt_Effect" Then
                    If ProfileLength <= 8 Then
                        For X = 0 To ProfileLength
                            Y = X+1
                            Child("Soil" & Y) = BrokenFam(X)
                            Child("Depth" & Y) = Elevations(X)
                        Next X
                    Else
                        For X = 0 To 8
                            Y = X+1
                            Child("Soil" & Y) = BrokenFam(X)
                            Child("Depth" & Y) = Elevations(X)
                        Next X
                    End If
                End If
            Next Child
```

```
        For Each Child In ob.ChildElements
            If Child.ElementType="AA_Dirt_Effect" Then
                Set OutEnt = ob.AddEntity
                Child.ScheduleEvent OutEnt,"ProcessEffect", 0.0001
            End If
        Next Child

        'Schedule the next observation and training of the net
        Set Ent1 = ob.AddEntity
        ob.ScheduleEvent Ent1,"Observe", ob("ObservationInterval")

    End Select

End Sub
```

```
Private Function FindElev(Northing1 As Double, Easting1 As Double, Elevation1 As Double, Northing2 As Double, Easting2 As Double, Elevation2 As Double, Northing3 As Double, Easting3 As Double, Elevation3 As Double, Northing As Double, Easting As Double) As Double

    Dim A As Double
    Dim B As Double
    Dim C As Double
Dim C1 As Double

Dim C2 As Double

Dim C3 As Double

Dim D As Double

Dim MyMatrix(3, 4) As Double

Dim TheMin As Double

Dim X As Integer

Dim Y As Integer


MyMatrix(1, 1) = Northing1

MyMatrix(1, 2) = Easting1

MyMatrix(1, 3) = Elevation1

MyMatrix(1, 4) = -1


MyMatrix(2, 1) = Northing2

MyMatrix(2, 2) = Easting2

MyMatrix(2, 3) = Elevation2

MyMatrix(2, 4) = -1


MyMatrix(3, 1) = Northing3

MyMatrix(3, 2) = Easting3

MyMatrix(3, 3) = Elevation3

MyMatrix(3, 4) = -1
```

```
C1 = MyMatrix(1, 1)
C2 = MyMatrix(2, 1)
C3 = MyMatrix(3, 1)

For X = 1 To 4
   MyMatrix(2, X) = MyMatrix(1, X) * C2 - MyMatrix(2, X) * C1
   MyMatrix(3, X) = MyMatrix(1, X) * C3 - MyMatrix(3, X) * C1
Next X

C2 = MyMatrix(2, 2)
C3 = MyMatrix(3, 2)

For X = 1 To 4
   MyMatrix(3, X) = MyMatrix(3, X) * C2 - MyMatrix(2, X) * C3
Next X

C1 = MyMatrix(1, 3)
C2 = MyMatrix(2, 3)
C3 = MyMatrix(3, 3)

For X = 1 To 4
   MyMatrix(1, X) = MyMatrix(1, X) * C3 - MyMatrix(3, X) * C1
```

```
    MyMatrix(2, X) = MyMatrix(2, X) * C3 - MyMatrix(3, X) * C2
Next X

C1 = MyMatrix(1, 2)
C2 = MyMatrix(2, 2)

For X = 1 To 4
    MyMatrix(1, X) = MyMatrix(1, X) * C2 - MyMatrix(2, X) * C1
Next X

C1 = MyMatrix(1, 1)
C2 = MyMatrix(2, 2)
C2 = MyMatrix(3, 3)


If C1 = 0 Then C1 = 0.0001
If C2 = 0 Then C2 = 0.0001
If C3 = 0 Then C3 = 0.0001

For X = 1 To 4
    MyMatrix(1, X) = MyMatrix(1, X) / C1
    MyMatrix(2, X) = MyMatrix(2, X) / C2
    MyMatrix(3, X) = MyMatrix(3, X) / C3
```

```
Next X


If MyMatrix(1, 1) = 0 Then MyMatrix(1, 1) = 0.0001

If MyMatrix(2, 2) = 0 Then MyMatrix(2, 2) = 0.0001

If MyMatrix(3, 3) = 0 Then MyMatrix(3, 3) = 0.0001



MyMatrix(1, 4) = -1 * MyMatrix(1, 4) / MyMatrix(1, 1)

MyMatrix(2, 4) = -1 * MyMatrix(2, 4) / MyMatrix(2, 2)

MyMatrix(3, 4) = -1 * MyMatrix(3, 4) / MyMatrix(3, 3)


If MyMatrix(1, 4) < MyMatrix(2, 4) Then

   If MyMatrix(1, 4) < MyMatrix(3, 4) Then TheMin = MyMatrix(1, 4)

End If


If MyMatrix(2, 4) < MyMatrix(1, 4) Then

   If MyMatrix(2, 4) < MyMatrix(3, 4) Then TheMin = MyMatrix(2, 4)

End If


If MyMatrix(3, 4) < MyMatrix(1, 4) Then

   If MyMatrix(3, 4) < MyMatrix(2, 4) Then TheMin = MyMatrix(3, 4)

End If
```

```
        If TheMin = 0 Then TheMin = 0.0001

        A = MyMatrix(1, 4) / TheMin
        B = MyMatrix(2, 4) / TheMin
        C = MyMatrix(3, 4) / TheMin
        D = 1 / TheMin

        If C = 0 Then C= 0.0001

        FindElev = (D - A * Northing - B * Easting) / C

End Function
```

A9.3.2 Condition Element Programming Code

```
Option Explicit
```

```
Public Function AA_Dirt_Condition_OnCreate(ob As CFCSim_ModelingElementInstance, x As Single, y As Single) As Boolean

        Dim Counter As Integer
        Dim Child As CFCSim_ModelingElementInstance
        Counter = 0
```

```
For Each Child In ob.Parent.ChildElements
        If Child.ElementType="AA_Dirt_Condition" Then
                Counter = Counter + 1
        End If
Next Child

If Counter = 0 Then
        AA_Dirt_Condition_OnCreate=True
        ob.OnCreate x,y,True
Else
        MessagePrompt "Prototype is only designed for one condition and effect
element per agent"
        AA_Dirt_Condition_OnCreate=False
End If

ob.AddAttribute "Northing","Northing",CFC_Numeric,CFC_Single,CFC_ReadWrite
ob.AddAttribute "Easting","Easting",CFC_Numeric,CFC_Single,CFC_ReadWrite
ob.AddAttribute                          "Radius","Viewing                    Radius
(m)",CFC_Numeric,CFC_Single,CFC_ReadWrite

ob("Northing") = 5929100
ob("Easting") = 37850
ob("Radius") = 1500
```

```
        ob.SetNumCoordinates 2
        ob.CoordinatesX(0)=x
        ob.CoordinatesY(0)=y
        ob.CoordinatesX(1)=x+80
        ob.CoordinatesY(1)=y+80

End Function
```

```
Public Sub AA_Dirt_Condition_OnDragDraw(ob As CFCSim_ModelingElementInstance)
        ob.OnDraw
End Sub
```

```
Public Sub AA_Dirt_Condition_OnDraw(ob As CFCSim_ModelingElementInstance)
        CDC.RenderPicture
"if.bmp",ob.CoordinatesX(0),ob.CoordinatesY(0),ob.CoordinatesX(1)-
ob.CoordinatesX(0),ob.CoordinatesY(1)-ob.CoordinatesY(0)

        If ob.Selected Then
                CDC.ChangeLineStyle CFC_DOT,1,RGB(255,0,0)
                CDC.Rectangle                    ob.CoordinatesX(0)-2,ob.CoordinatesY(0)-
2,ob.CoordinatesX(1)+2,ob.CoordinatesY(1)+2
```

```
        End If

        CDC.ChangeLineStyle CFC_SOLID,1,RGB(0,0,0)

End Sub
```

```
Public Sub AA_Dirt_Condition_OnSimulationInitialize(ob As
CFCSim_ModelingElementInstance)
        ob.AddEvent "CheckModel",True
End Sub
```

```
Public Sub AA_Dirt_Condition_OnSimulationProcessEvent(ob As
CFCSim_ModelingElementInstance, MyEvent As String, Entity As CFCSim_Entity)
Dim x As String
Dim y As String
Dim Z As String

        Select Case MyEvent
                Case "CheckRule"
                                x = ob("Northing")
                                y = ob("Easting")
                                Z = ob("Radius")
        End Select
End Sub
```

A9.3.3 Effect Element Programming Code

```
Option Explicit
```

```
Public Function AA_Dirt_Effect_OnCreate(ob As CFCSim_ModelingElementInstance, X As Single, y As Single) As Boolean

    Dim Counter As Integer
    Dim Child As CFCSim_ModelingElementInstance
    Counter = 0
    For Each Child In ob.Parent.ChildElements
        If Child.ElementType="AA_Dirt_Effect" Then
            Counter = Counter + 1
        End If
    Next Child

    If Counter = 0 Then
        AA_Dirt_Effect_OnCreate=True
        ob.OnCreate X,y,True
    Else
```

```
        MessagePrompt "Prototype is only designed for one condition and effect element per agent"
        AA_Dirt_Effect_OnCreate=False
    End If



    ob.AddAttribute "Soil1","Soil Layer 1 Type",CFC_Text,CFC_Single,CFC_ReadWrite
    ob.AddAttribute "Depth1","Soil Layer 1 StartDepth",CFC_Numeric,CFC_Single,CFC_ReadWrite
    ob.AddAttribute "Soil2","Soil Layer 2 Type",CFC_Text,CFC_Single,CFC_ReadWrite
    ob.AddAttribute "Depth2","Soil Layer 2 StartDepth",CFC_Numeric,CFC_Single,CFC_ReadWrite
    ob.AddAttribute "Soil3","Soil Layer 3 Type",CFC_Text,CFC_Single,CFC_ReadWrite
    ob.AddAttribute "Depth3","Soil Layer 3 StartDepth",CFC_Numeric,CFC_Single,CFC_ReadWrite
    ob.AddAttribute "Soil4","Soil Layer 4 Type",CFC_Text,CFC_Single,CFC_ReadWrite
    ob.AddAttribute "Depth4","Soil Layer 4 StartDepth",CFC_Numeric,CFC_Single,CFC_ReadWrite
```

```
        ob.AddAttribute "Soil5","Soil Layer 5 Type",CFC_Text,CFC_Single,CFC_ReadWrite
        ob.AddAttribute "Depth5","Soil Layer 5 StartDepth",CFC_Numeric,CFC_Single,CFC_ReadWrite
        ob.AddAttribute "Soil6","Soil Layer 6 Type",CFC_Text,CFC_Single,CFC_ReadWrite
        ob.AddAttribute "Depth6","Soil Layer 6 StartDepth",CFC_Numeric,CFC_Single,CFC_ReadWrite
        ob.AddAttribute "Soil7","Soil Layer 7 Type",CFC_Text,CFC_Single,CFC_ReadWrite
        ob.AddAttribute "Depth7","Soil Layer 7 StartDepth",CFC_Numeric,CFC_Single,CFC_ReadWrite
        ob.AddAttribute "Soil8","Soil Layer 8 Type",CFC_Text,CFC_Single,CFC_ReadWrite
        ob.AddAttribute "Depth8","Soil Layer 8 StartDepth",CFC_Numeric,CFC_Single,CFC_ReadWrite
        ob.AddAttribute "Soil9","Soil Layer 9 Type",CFC_Text,CFC_Single,CFC_ReadWrite
        ob.AddAttribute "Depth9","Soil Layer 9 StartDepth",CFC_Numeric,CFC_Single,CFC_ReadWrite
        ob.AddAttribute "Evaluation","Evaluation Value - if reference is needed to model component",CFC_Text,CFC_Single,CFC_ReadWrite
        ob.AddAttribute "Effect","Effect (Linked
Value)",CFC_Text,CFC_Single,CFC_ReadWrite

        ob.SetNumCoordinates 2
        ob.CoordinatesX(0)=X
        ob.CoordinatesY(0)=y
        ob.CoordinatesX(1)=X+80
        ob.CoordinatesY(1)=y+80
End Function
```

```
Public Sub AA_Dirt_Effect_OnDragDraw(ob As CFCSim_ModelingElementInstance)
        ob.OnDraw
End Sub
```

```
Public Sub AA_Dirt_Effect_OnDraw(ob As CFCSim_ModelingElementInstance)
        CDC.RenderPicture
"then.bmp",ob.CoordinatesX(0),ob.CoordinatesY(0),ob.CoordinatesX(1)-
ob.CoordinatesX(0),ob.CoordinatesY(1)-ob.CoordinatesY(0)

        If ob.Selected Then
                CDC.ChangeLineStyle CFC_DOT,1,RGB(255,0,0)
                CDC.Rectangle ob.CoordinatesX(0)-2,ob.CoordinatesY(0)-
2,ob.CoordinatesX(1)+2,ob.CoordinatesY(1)+2
```

```
        End If

        CDC.ChangeLineStyle CFC_SOLID,1,RGB(0,0,0)

End Sub
```

```
Public Sub AA_Dirt_Effect_OnSimulationInitialize(ob As CFCSim_ModelingElementInstance)
        ob.AddEvent "ProcessEffect",True
End Sub
```

```
Public Sub AA_Dirt_Effect_OnSimulationProcessEvent(ob As CFCSim_ModelingElementInstance, MyEvent As String, Entity As CFCSim_Entity)
        Dim X As String

        Select Case MyEvent
        Case "ProcessEffect"
                X = ob("Effect")
        End Select

End Sub
```

A9.4 Rule Based Agent

The following subsections provide the information pertaining to the construction of the modeling elements that make up the rule based agent presented in this research.

A9.4.1 Parent Agent Element Programming Code

```
Option Explicit
```

```
Public Function AA_Heur_OnCreate(ob As CFCSim_ModelingElementInstance, x As Single, y As Single) As Boolean
        AA_Heur_OnCreate=True
        ob.OnCreate x,y,True

        ob.AddAttribute "FirstObs","Time of First Observation",CFC_Numeric,CFC_Single,CFC_ReadWrite
        ob.AddAttribute "ObservationInterval","Observation Interval",CFC_Numeric,CFC_Single,CFC_ReadWrite

        ob.Attr("ObservationInterval") = 100
        ob.Attr("FirstObs") = 0

        ob.SetNumCoordinates 2
        ob.CoordinatesX(0)=x
        ob.CoordinatesY(0)=y
        ob.CoordinatesX(1)=x+70
        ob.CoordinatesY(1)=y+49
End Function
```

```
Public Sub AA_Heur_OnDragDraw(ob As CFCSim_ModelingElementInstance)
        ob.OnDraw
End Sub
```

```
Public Sub AA_Heur_OnDraw(ob As CFCSim_ModelingElementInstance)
        CDC.RenderPicture
"harrys1.bmp",ob.CoordinatesX(0),ob.CoordinatesY(0),ob.CoordinatesX(1)-
ob.CoordinatesX(0),ob.CoordinatesY(1)-ob.CoordinatesY(0)

        If ob.Selected Then
                CDC.ChangeLineStyle CFC_DOT,1,RGB(255,0,0)
                CDC.Rectangle                    ob.CoordinatesX(0)-2,ob.CoordinatesY(0)-
2,ob.CoordinatesX(1)+2,ob.CoordinatesY(1)+2
        End If

        CDC.ChangeLineStyle CFC_SOLID,1,RGB(0,0,0)
End Sub
```

```
Public Sub AA_Heur_OnSimulationInitialize(ob As CFCSim_ModelingElementInstance)

        ob.AddEvent "Observe",True

End Sub
```

```
Public Sub AA_Heur_OnSimulationInitializeRun(ob As
CFCSim_ModelingElementInstance, RunNum As Integer)
        Dim ent As CFCSim_Entity
        Set ent = ob.AddEntity

        ob.ScheduleEvent ent,"Observe",ob("FirstObs")

End Sub
```

```
Public Sub AA_Heur_OnSimulationProcessEvent(ob As
CFCSim_ModelingElementInstance, MyEvent As String, Entity As CFCSim_Entity)

        Dim Child As CFCSim_ModelingElementInstance
        Dim OutEnt As CFCSim_Entity
        Dim OutEnt1 As CFCSim_Entity
        Dim SecondOutEnt As CFCSim_Entity

        Set OutEnt = ob.AddEntity
        Set SecondOutEnt = ob.AddEntity


        Select Case MyEvent

        Case "Observe"


                For Each Child In ob.ChildElements

                        If Child.ElementType="AA_Heur_Condition" Then

                                Set OutEnt = ob.AddEntity

                                Child.ScheduleEvent OutEnt,"CheckModel", 0

                        End If

                Next Child


                For Each Child In ob.ChildElements

                        If Child.ElementType="AA_Heur_Effect" Then

                                Set OutEnt1 = ob.AddEntity

                                Child.ScheduleEvent OutEnt1,"ProcessEffect", 0.0001

                        End If

                Next Child


                ob.ScheduleEvent SecondOutEnt,"Observe", ob("ObservationInterval")


        End Select

End Sub
```

A9.4.2 Condition Element Programming Code

```
Option Explicit
```

```
Public Function AA_Heur_Condition_OnCreate(ob As CFCSim_ModelingElementInstance, x As Single, y As Single) As Boolean
        AA_Heur_Condition_OnCreate=True
        ob.OnCreate x,y,True

        ob.AddAttribute "ObsValue","Input Variable To Observe (Linked Value)",CFC_Text,CFC_Single,CFC_ReadWrite

        ob.SetNumCoordinates 2
        ob.CoordinatesX(0)=x
        ob.CoordinatesY(0)=y
        ob.CoordinatesX(1)=x+80
        ob.CoordinatesY(1)=y+80
End Function
```

```
Public Sub AA_Heur_Condition_OnDragDraw(ob As CFCSim_ModelingElementInstance)
        ob.OnDraw
End Sub
```

```
Public Sub AA_Heur_Condition_OnDraw(ob As CFCSim_ModelingElementInstance)
        CDC.RenderPicture
"if.bmp",ob.CoordinatesX(0),ob.CoordinatesY(0),ob.CoordinatesX(1)-
ob.CoordinatesX(0),ob.CoordinatesY(1)-ob.CoordinatesY(0)

        If ob.Selected Then
                CDC.ChangeLineStyle CFC_DOT,1,RGB(255,0,0)
                CDC.Rectangle ob.CoordinatesX(0)-2,ob.CoordinatesY(0)-
2,ob.CoordinatesX(1)+2,ob.CoordinatesY(1)+2
        End If

        CDC.ChangeLineStyle CFC_SOLID,1,RGB(0,0,0)
End Sub
```

```
Public Sub AA_Heur_Condition_OnSimulationInitialize(ob As
CFCSim_ModelingElementInstance)
        ob.AddEvent "CheckModel",True
End Sub
```

```
Public Sub AA_Heur_Condition_OnSimulationProcessEvent(ob As
CFCSim_ModelingElementInstance, MyEvent As String, Entity As CFCSim_Entity)
        Dim x As String
        Select Case MyEvent
                Case "CheckModel"
                                x = ob("ObsValue")
        End Select
End Sub
```

A9.4.3 Effect Element Programming Code

```
Option Explicit
```

```
Public Function AA_Heur_Effect_OnCreate(ob As CFCSim_ModelingElementInstance, X
As Single, y As Single) As Boolean
        AA_Heur_Effect_OnCreate=True
        ob.OnCreate X,y,True

        ob.AddAttribute "Evaluation","Evaluation Value - if reference is needed to model
component",CFC_Text,CFC_Single,CFC_ReadWrite
        ob.AddAttribute "Effect","Effect (Linked
Value)",CFC_Text,CFC_Single,CFC_ReadWrite

        ob.SetNumCoordinates 2
        ob.CoordinatesX(0)=X
        ob.CoordinatesY(0)=y
        ob.CoordinatesX(1)=X+80

        ob.CoordinatesY(1)=y+80

End Function
```

```
Public Sub AA_Heur_Effect_OnDragDraw(ob As CFCSim_ModelingElementInstance)

        ob.OnDraw

End Sub
```

```
Public Sub AA_Heur_Effect_OnDraw(ob As CFCSim_ModelingElementInstance)

        CDC.RenderPicture

"then.bmp",ob.CoordinatesX(0),ob.CoordinatesY(0),ob.CoordinatesX(1)-

ob.CoordinatesX(0),ob.CoordinatesY(1)-ob.CoordinatesY(0)


        If ob.Selected Then

                CDC.ChangeLineStyle CFC_DOT,1,RGB(255,0,0)

                CDC.Rectangle                    ob.CoordinatesX(0)-2,ob.CoordinatesY(0)-

2,ob.CoordinatesX(1)+2,ob.CoordinatesY(1)+2

        End If


        CDC.ChangeLineStyle CFC_SOLID,1,RGB(0,0,0)

End Sub
```

```
Public Sub AA_Heur_Effect_OnSimulationInitialize(ob As
CFCSim_ModelingElementInstance)
        ob.AddEvent "ProcessEffect",True
End Sub
```

```
Public Sub AA_Heur_Effect_OnSimulationProcessEvent(ob As
CFCSim_ModelingElementInstance, MyEvent As String, Entity As CFCSim_Entity)
        Dim X As String

        Select Case MyEvent
        Case "ProcessEffect"
                X = ob("Effect")
        End Select

End Sub
```